ラジオライフ別冊

ラジオライフ別冊「バックアップ活用テクニック2」 昭和60年11月15日発行

PC/FM/X1/ファミコンGAMEハッカーに贈る!!

バックアップ活用テクニック

パート2



- FM,PC/DISKコピーツール集
- ファミコン初公開解析テクニック
- X1完全解剖と強力Back UPソフト
- MSXハード・ソフトBack UP法

アフターサポート万全！

EXPERTが1枚あれば、他のコピーツールは不要になる！

Analyzer & Copy Construction.

FM-7・77 DISK COPY TOOL

EXPERT FM

Version 3.1

PC-8801/mkII/SR DISK COPY TOOL

EXPERT 88

新発売

特徴

I．強力無比なオートマチックバックアップモードによりコンストラクションモードの使用を、可能な限り少なくしています。

II．万一、オートマチックでコピーできないソフトがあっても、コンストラクションモードによりカバーできます。このモードでは、FM、PCに使用されているFDCの関係上、本来バックアップできないものなどに対応します。

III．高性能ディスクアナライザーを内蔵していますので、オリジナル拡張BASICコマンドと併せてコンストラクションファイルを作成するのに大変役立ちます。

IV．アフターサポートは、コンストラクションファイルサービスに加え、情報誌「EXPERT-NOTE」により万全を期す体制を取っております。

拡張BASICコマンド

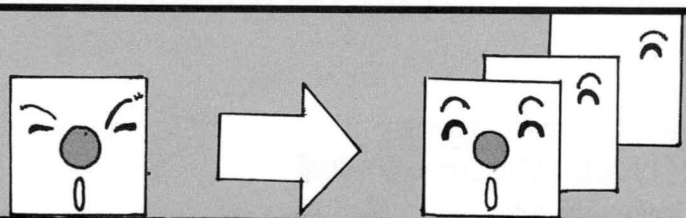
EXPERTシリーズでは各機種のBASICにコマンドを拡張して、誰にでも簡単にディスクを操作できるようにしています。ですからマシン語がわからなくても特殊なフォーマットをBASICで取り扱うことができます。拡張コマンドの取り扱いについては付属のマニュアルに詳しく解説しています。

FM-7・77版
PC-8801/mkII/SR版共
定価12,800円
好評発売中

SOFTPAL
ソフパル

〒556
大阪市浪速区日本橋4丁目7-22
TEL 06(644)3782

お求めは全国の有名パソコンショップで。通信販売も受け賜っております。御注文の際は、住所・氏名・電話番号・使用機種を明記して、現金書留にてお申し込み下さい。(送料無料)



BABY MAKER

最強の バックアップ・プログラム “ベビーメーカー”

PC-9801/E/F/M/U

ばん・ばん・とれる どん・どん・おちる

一枚のマスターから何枚でもバックアップOK
オートも最強・パラメーターで書き換えもOK

8インチ、5インチ、2DD、3.5インチ 各¥14,800
PC-8801/mkII/SR用もあります。 ¥12,800
プロテクトマスター¥4,800、N88ディスクアナライザー¥6,800

[PC-9801/E/F/M/U
増設 RAM.CARD

やすい!

256K→¥19,800　512K→¥32,000

ノーブランド ディスケット
5インチ 2D

1枚¥190
何枚でも
送料¥1,000

マイコンシステム

☆お求めは全国有名マイコンショップ
☆直接当社でも販売致します。現金又は、書留または銀行振込をご利用下さい。
☆振込先：太陽神戸銀行　高田馬場支店
普通 3179582（御注文金額5000円以上送料無料、5000円以下は〒250必要）

(お問い合せ)〒171 東京都豊島区高田 3－14－24
(10時～7時) ハイライフ高田馬場 102号室
☎03(981)0563

バックアップ活用テクニック❷ 目次

読者スペシャルプレゼント



DISKアナライザーといえば、まず最初に浮かぶのが〝RATS＆STAR〞です。専用の簡易言語を駆使し、バッタバッタとプロテクトをなぎたおすには、抜群の威力を発揮します。

ところで、Atarixから、この〝RATS＆STAR〞のPC−8801（5インチ）用を2本、FM−7（5インチ）用を2本、FM−77（3.5インチ）用を2本読者の皆さんに提供していただくことになりました。ご希望の方は、三才ブックス『RATS＆STAR』係まで、住所、氏名、年齢、本書についての感想を葉書に明記の上お送りください。締め切りは12月末日(消印有効)です。

第1章
【FM/PC-8801】
DISK
コピー・ツール集



オートモードDISKバックアップ・ツール

```
PC-8801/mk2/SR   By Gakincho
Disk Backup Tool Version 1.0

READ DRIVE(1)] 1  WRITE DRIVE(2)] 2

START TRACK(00)] 00  END TRACK(79)] 79

FORMAT PARAMETER INDICATE? No

Copy from Drive 1 to Drive 2
Copy from Track 0 to Track 79
Copying Track 0 Copying Track 1
Copying Track 2 Copying Track 3
Copying Track 4 Copying Track 5
Copying Track 6 Copying Track 7
Copying Track 8 Copying Track 9
Copying Track 10 Copying Track 11
```

PC—8801/mkII/SR

"オニャンコN88"



BY
ガキンチョ

●はじめに

この記事を読まれて、なぜ今頃オート・コピーなのだろうか？　と思われた方がいらしゃるでしょう。確かに現在のプロテクトは高度になり、オート・コピーでは取れないソフトが多数あります。

ですから当然、アナライザーを使ってプロテクトの解読もしなければなりません。

アナライザーを使う場合には、基本的な予備知識さえあれば、後はアナライザー操作法さえ分かれば誰にでも使えるようになります。

しかし、書き換えを行なう場合には、もっと高度な知識が必要です。また、バックアップに関してのカンも持ってしかも、マシン語を知っておかなければなりません。

例えば、ＦＤＣ（フロッピー・ディスク・コントローラ）に関する事柄は、必ず理解しておかなくてはお話しになりません。このようにプロテクト技術が発達した現在ではソフトの書き換えなしでは、プロテクトは外せないでしょう。

そこで登場するのが､このプログラムです。何故かというと、まずこのプログラムを皆さんで解読し知識を身に付けて欲しいからです。

これからオリジナルのコピー・ツールを作られる方なら本プログラムを参考にするのも良いでしょうし、上級者の皆さんならこれを改良することもできます。

しかし、プログラムを解読する際には、プログラムを良く見てください。プログラムをいじられる方達のために、なるべく理解し易く作ってあるからです。(上級者のなかには無駄なコトをしているように思われるかもしれませんが)

●プログラムの入力

本プログラムはN88-DISK BASICで動きます。(バージョンは何でもかまいません)BASICとマシン語で構成しており、どちらを先に打ち込んでもかまいません。BASICプログラムのファイル・ネームは自分で好きなように決めてSAVEしてください。

マシン語プログラムは打ち込んだ後､SAVEする際のファイル・ネームは〝back up 88″にしておいてください。BASICをRUNさせた時、マシン語がないと自動的にロードを開始します。

〈リスト1〉BASIC部




```
1000 '****************************************
1010 '*                                      *
1020 '* PC-8801/mk2/SR Disk Backup Program *
1030 '*                                      *
1040 '* Version 1.0    AUGUST/27/1985        *
1050 '*                                      *
1060 '* Copyright(c) By ｵﾆｬﾝｺ SOFT           *
1070 '*                                      *
1080 '* PROGRAMMED BY GAKINCHO               *
1090 '*                                      *
1100 '****************************************
1110 '***** INITIALIZE *****
1120 WIDTH 40,20:SCREEN 0:CONSOLE 0,25,0,1:COLOR 7:CLEAR 300,&HBFFF
1130 IF PEEK(&HC000)<>&HD7 THEN BLOAD"backup88":ADD=&HCBFF:CALL ADD
1140 PRINT"PC-8801/mk2/SR   By Gakincho
1150 PRINT"Disk Backup Tool Version 1.0
1160 '***** READ/WRITE DRIVE No. INPUT *****
1170 LOCATE 0,3:PRINT"READ DRIVE(1)]":LOCATE 15,3:RD$=INPUT$(1):PRINT RD$:IF RD$
=CHR$(13) THEN LOCATE 14,3:RD=1:PRINT RD ELSE RD=VAL(RD$):IF RD<1 OR 4<RD THEN 1
170
1180 LOCATE 18,3:PRINT"WRITE DRIVE(2)]":LOCATE 34,3:WD$=INPUT$(1):PRINT WD$:IF W
D$=CHR$(13) THEN LOCATE 33,3:WD=2:PRINT WD ELSE WD=VAL(WD$):IF WD<1 OR 4<WD THEN
 1180
1190 '***** START/END TRACK No INPUT *****
1200 PRINT:LINE INPUT"START TRACK(00)] ",ST$:IF ST$="" THEN ST=0:LOCATE 17,5:PRI
NT"00" ELSE ST=VAL(ST$)
1210 IF ST<0 OR 79<ST THEN BEEP:GOTO 1200
1220 LOCATE 21,5:LINE INPUT"END TRACK(79)] ",ET$:IF ET$="" THEN ET=79:LOCATE 36,
5:PRINT"79" ELSE ET=VAL(ET$)
1230 IF ET<0 OR 79<ET THEN BEEP:GOTO 1220
1240 '***** FORMAT PARAMETER INDICATE SWITCH INPUT *****
1250 PRINT:PRINT"FORMAT PARAMETER INDICATE?"
1260 LOCATE 27,7:S$=INPUT$(1):S=INSTR("YyﾝNnﾐ"+CHR$(13),S$):IF S=0 THEN 1260
1270 IF S>3 THEN PRINT"No":FPFLAG=0 ELSE PRINT"Yes":FPFLAG=-1
1280 '***** INPUT PARAMETER INDICATE *****
1290 PRINT:PRINT"Copy from Drive";RD;"to Drive";WD
1300 PRINT"Copy from Track";ST;"to Track";ET
1310 '***** MAIN *****
1315 CONSOLE 11,25
1320 FOR L=ST TO ET
1330 D=&HD:GOSUB *COMMND:D=&H70:GOSUB *SHAND:D=&H0:GOSUB *SHAND 'DISK SIDE PROGR
AM EXECUTE
1340 PRINT"Copying Track";L; 'TRACK NUMBER INDICATE
1350 D=RD-1:GOSUB *SHAND:D=WD-1:GOSUB *SHAND:D=L:GOSUB *SHAND 'PARAMETER SEND
1360 GOSUB *AHAND:E=INSTR("UHwd",CHR$(D)):IF E=1 THEN GOSUB 1400 ELSE IF E>1 THE
N PRINT"Disk I/O Error ｶﾞ ﾊｯｾｲｼﾏｼﾀ STOP-RESETｦｼﾃ ﾓｳｲｯｶｲ RUNｼﾃｸﾀﾞｻｲ":GOTO 1640 'E
RROR CHECK
1370 GOSUB *AHAND:LENGTH=D:GOSUB *AHAND:FDATA=D:GOSUB *AHAND:GPL=D:GOSUB *AHAND:
SC=D:IF FPFLAG THEN PRINT"N ";HEX$(LENGTH);" SC ";HEX$(SC);" GPL ";HEX$(GPL);" D
 ";HEX$(FDATA);" " 'FORMAT PARAMETER INDICATE
1380 NEXT L
1390 CONSOLE 0,25:GOTO 1160 'DRIVE No INPUT JUMP
1400 PRINT"UNFORMAT":RETURN 'UNFORMAT INDICATE
1410 CONSOLE 0,25:GOTO 1160 'DRIVE No INPUT JUMP
1420 '***** SOURCE HAND SHAKE *****
1430 *COMMND
1440 OUT &HFF,&HF
1450 *SHAND
1460 WAIT &HFE,&H2,&H0
1470 OUT &HFF,&HE
1480 OUT &HFD,D
1490 OUT &HFF,&H9
1500 WAIT &HFE,&H4,&H0
1510 OUT &HFF,&H8
1520 WAIT &HFE,&H4,&H4
1530 RETURN
1540 '***** ACCEPTER HAND SHAKE *****
1550 *AHAND
1560 OUT &HFF,&HB
1570 WAIT &HFE,&H1,&H0
1580 OUT &HFF,&HA
1590 D=INP(&HFC)
1600 OUT &HFF,&HD
1610 WAIT &HFE,&H1,&H1
1620 OUT &HFF,&HC
1630 RETURN
1640 '***** Disk I/O ERROR *****
1650 END
```

〈リスト2〉マシン語部

```
ADRS   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  SUM  [0123456789ABCDEF]
-------------------------------------------------------------------------------
C000 : D7 32 00 64 D7 32 01 64:D7 32 02 64 3A 00 64 4F : 37  |ﾗ2 dﾗ2 dﾗ2 d: dO|
C010 : 3A 02 64 57 CD AA 01 38:0F 3A 01 64 4F 3A 02 64 : 44  |: dW^ｪ 8 : dO: d|
C020 : 57 CD AA 01 38 02 18 08:3E 48 32 2E 64 C3 8A 7A : 3A  |W^ｪ 8   >H2.dﾃ z|
C030 : 3A 00 64 4F 3A 02 64 57:06 30 21 00 60 22 07 64 : 28  |: dO: dW 0! `" d|
C040 : 21 00 62 22 09 64 3E 0A:CD A4 02 CD 0E 04 E7 CD : 60  |! b" d> ^､ ^  ◤^|
C050 : 0D 7B CD 0D 7B CD 0D 7B:CD 0D 7B CD 0D 7B CD 0D : B6  | {^ {^ {^ {^ {^ |
C060 : 7B CD 0D 7B FD 21 00 00:3E 4A CD A4 02 CD 0E 04 : C8  |{^ { !  >J^､ ^  |
C070 : E7 CD 01 7B CD 01 7B 32:0E 7F CD 01 7B 2A 07 64 : 16  |◤^ {^ {2  ^ {* d|
C080 : CD 01 7B 77 23 CD 01 7B:77 23 CD 01 7B 77 23 CD : 76  |^ {w#^ {w#^ {w#^|
C090 : 01 7B 77 23 22 07 64 3A:0E 7F E6 05 C2 A6 7A D5 : 0C  | {w#" d:  ◥ ﾂｦzﾕ|
C0A0 : FD E5 D1 2A 09 64 73 23:72 23 22 09 64 D1 10 B4 : 99  | ◣ﾑ* ds#r#" dﾑ ｴ|
C0B0 : 3A 02 60 32 FD 63 21 02:60 16 01 23 23 23 23 7E : D2  |: `2 c! `  ####~|
C0C0 : 5F 3A FD 63 BB 28 03 14:18 F1 7A 32 FC 63 3D 32 : 76  | : cｻ(   円z2 c=2|
C0D0 : FE 63 21 00 00 22 E0 63:11 7F 3B 2A 02 62 EB ED : 18  | c!  "=c  ;* b♣○|
C0E0 : 52 EB DA 88 71 21 00 62:4E 23 46 23 ED 5B E0 63 : F8  |R♣ﾚ q! bN#F#○[=c|
C0F0 : E5 60 69 19 22 E0 63 E1:3A FE 63 3D 32 FE 63 FE : 76  |◣`i "=cﾄ: c=2 c |
-------------------------------------------------------------------------------
 SUM : CB 61 33 2A FD 19 83 46 18 CA A1 23 C6 C4 FB 27    BA

ADRS   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  SUM  [0123456789ABCDEF]
-------------------------------------------------------------------------------
C100 : 00 28 02 18 E3 2A E0 63:11 7F 3B EB ED 52 EB DA : 4C  | (  ｲ*=c  ;♣○R♣ﾚ|
C110 : 88 71 21 00 00 22 E0 63:3E 01 32 E2 63 2A 00 62 : C1  | q!  "=c> 2ｷc* b|
C120 : ED 5B 02 62 19 11 7F 3B:EB ED 52 EB 38 52 3E FF : 6C  |○[ b   ;♣○R♣8R> |
C130 : 32 E2 63 21 00 62 4E 23:46 23 3A E2 63 3C 32 E2 : A3  |2ｷc! bN#F#:ｷc<2ｷ|
C140 : 63 ED 5B E0 63 E5 60 69:19 22 E0 63 E1 E5 C5 2A : CF  |c○[=c◣`i "=cﾄ◣ﾅ*|
C150 : E0 63 11 7F 3B EB ED 52:EB 38 25 C1 E1 5E 23 56 : F9  |=c  ;♣○R♣8%ﾁﾄ^#V|
C160 : 23 3A E2 63 3C 32 E2 63:E5 2A E0 63 19 22 E0 63 : 25  |#:ｷc<2ｷc◣*=c "=c|
C170 : E1 ED 4B E0 63 E5 21 7F:3B ED 42 38 03 E1 18 B6 : 35  |ﾄ○K=c◣! ;○B8 ﾄ ｶ|
C180 : 3A E2 63 32 0D 64 18 08:3A FC 63 32 0D 64 18 00 : 96  |:ｷc2 d  : c2 d  |
C190 : 3A 0D 64 32 0E 64 21 FF:5F 22 0F 64 21 5F 62 22 : 67  |: d2 d! _" d!_b"|
C1A0 : 11 64 2A 0F 64 23 23 23:23 22 0F 64 7E 2A 11 64 : 50  | d* d####" d~* d|
C1B0 : 23 22 11 64 FE 00 28 28:FE 01 28 24 FE 02 28 20 : 9B  |#" d  ((  ($  ( |
C1C0 : FE 03 28 1C FE 04 28 18:FE 05 28 14 FE 06 28 10 : 02  |  (   (   (   ( |
C1D0 : 3E 00 77 3A 0E 64 3D 32:0E 64 FE 00 20 C4 18 0E : 4A  |> w: d=2 d   ﾄ  |
C1E0 : 3E FF 77 3A 0E 64 3D 32:0E 64 FE 00 20 B4 06 01 : 1A  |> w: d=2 d   ｴ  |
C1F0 : 78 32 15 64 CD D0 76 21:5F 62 23 22 11 64 E5 21 : D8  |x2 d^ﾐv!_b#" d◣!|
-------------------------------------------------------------------------------
 SUM : 88 F6 4E 08 9D 2D 79 B0 D7 71 10 AD C2 21 19 9C    64

ADRS   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  SUM  [0123456789ABCDEF]
-------------------------------------------------------------------------------
C200 : 30 62 22 13 64 21 00 60:22 07 64 E1 7E FE FF 28 : BD  |0b" d! `" dﾄ~  (|
C210 : 29 3A 15 64 47 3A 0D 64:B8 CA 63 73 04 78 32 15 : E9  |): dG: dｸﾊcs x2 |
C220 : 64 23 22 11 64 E5 21 30:62 23 22 13 64 21 00 60 : F3  |d#" d◣!0b#" d! `|
C230 : 23 23 23 23 22 07 64 E1:18 D2 CD 91 72 3A 15 64 : 67  |####" dﾄ ﾒ^ r: d|
C240 : 47 3A 0D 64 B8 CA 63 73:3A 15 64 3C 32 15 64 2A : 0E  |G: dｸﾊcs: d<2 d*|
C250 : 11 64 7E FE FF 28 2A 3A:15 64 47 3A 0D 64 B8 CA : 69  | d~  (*: dG: dｸﾊ|
C260 : 63 73 04 78 32 15 64 2A:11 64 23 22 11 64 2A 13 : 93  |cs x2 d* d#" d* |
C270 : 64 23 22 13 64 2A 07 64:23 23 23 23 22 07 64 18 : E6  |d#" d* d####" d |
C280 : CE CD 91 72 3A 15 64 47:3A 0D 64 B8 CA 63 73 18 : B3  |ﾎ^ r: dG: dｸﾊcs |
C290 : B7 3A 00 64 4F 3A 02 64:57 3E 46 CD A4 02 CD 0E : 6D  |ｷ: dO: dW>F^､ ^ |
C2A0 : 04 E7 2A 07 64 7E E7 23:7E E7 23 7E E7 23 7E E7 : 7D  | ◤* d~◤#~◤#~◤#~◤|
C2B0 : CD 2F 73 23 22 07 64 3E:01 E7 3E 01 E7 3E FF E7 : 8F  |^/s#" d> ◤> ◤> ◤|
C2C0 : 21 00 40 FB 76 DB FA E6:20 28 09 DB FB 77 23 0B : 59  |! @ v□ ◥ ( □ w# |
C2D0 : 79 B0 20 EF DB F8 FB 76:F3 CD 63 02 2A 13 64 3A : 7C  |y- ＼□ v月^c * d:|
C2E0 : 0E 7F FE 20 28 1B FE 01:28 2E 3A 0F 7F FE 40 20 : 69  |    (   (.:   @ |
C2F0 : 08 3E 04 77 23 22 13 64:C9 3E 05 77 23 22 13 64 : BC  | > w#" d/> w#" d|
-------------------------------------------------------------------------------
 SUM : 05 A0 BD 19 29 5C 41 DD EB 40 5D 1A CD 25 87 DD    16
```

```
ADRS    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [0123456789ABCDEF]
-----------------------------------------------------------------------------------
C300 :  C9 3A 0F 7F FE 20 20 08:3E 01 77 23 22 13 64 C9 :  12   |/:        > w#" d/|
C310 :  3E 00 77 23 22 13 64 C9:3A 0F 7F FE 01 20 08 3E :  67   |> w#" d/:        >|
C320 :  03 77 23 22 13 64 C9 3E:02 77 23 22 13 64 C9 FE :  39   | w#" d/> w#" d/  |
C330 :  00 28 18 FE 01 28 18 FE:02 28 18 FE 03 28 18 FE :  FE   | (   (   (   (   |
C340 :  04 28 18 FE 05 28 18 01:00 20 C9 01 80 00 C9 01 :  BC   | (   (    /    / |
C350 :  00 01 C9 01 00 02 C9 01:00 04 C9 01 00 08 C9 01 :  37   |  /   /   /    / |
C360 :  00 10 C9 21 30 62 22 13:64 21 00 60 22 07 64 3A :  6D   |  /!0b" d! `" d:|
C370 :  0D 64 32 1F 64 2A 13 64:7E FE 01 28 1F 3A 1F 64 :  48   | d2 d* d~  ( : d|
C380 :  3D 32 1F 64 FE 00 CA 68:74 2A 13 64 23 22 13 64 :  F3   |=2 d  ハht* d#" d|
C390 :  2A 07 64 23 23 23 23 22:07 64 18 D9 3A 00 64 4F :  8C   |* d####" d ル: dO|
C3A0 :  3A 02 64 57 3E 46 CD A4:02 CD 0E 04 E7 2A 07 64 :  49   |: dW>Fヘ、 ヘ  ◤* d|
C3B0 :  7E E7 23 7E E7 23 7E E7:23 7E E7 CD 2F 73 23 22 :  B1   |~◤#~◤#~◤#~◤ヘ/s#"|
C3C0 :  07 64 3E 01 E7 3E 01 E7:3E FF E7 21 00 40 FB 76 :  AD   | d> ◤> ◤> ◤! @ v|
C3D0 :  DB FA E6 20 28 09 DB FB:77 23 0B 79 B0 20 EF DB :  9A   |'フ ◥ ( □ w# y- ＼□|
C3E0 :  F8 FB 76 F3 CD 63 02 01:00 00 21 00 40 7E 32 16 :  B6   |  v月ヘc     ! @~2 |
C3F0 :  64 23 03 BE 28 F7 CD 17:74 3A 17 64 CD 2F 73 21 :  04   |d# セ(秒ヘ t: dヘ/s!|
-----------------------------------------------------------------------------------
 SUM :  78 14 44 2F 17 A2 5E 95 27 27 0E D7 2A D4 92 64    D2

ADRS    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [0123456789ABCDEF]
-----------------------------------------------------------------------------------
C400 :  00 40 09 23 23 06 00 7E:FE 00 28 04 04 23 18 F7 :  73   | @ ##  ~   (  # 秒|
C410 :  78 32 18 64 C3 B7 75 3E:80 B9 20 05 3E 00 C3 56 :  08   |x2 dテキu> ケ . > テV|
C420 :  74 3E 01 B8 20 05 3E 01:C3 56 74 3E 02 B8 20 05 :  79   |t> ク  > テVt> ク  |
C430 :  3E 02 C3 56 74 3E 04 B8:20 05 3E 03 C3 56 74 3E :  F8   |> テVt> ク  > テVt>|
C440 :  08 B8 20 05 3E 04 C3 56:74 3E 10 B8 20 05 3E 05 :  22   | ク  > テVt> ク  > |
C450 :  C3 56 74 C3 56 74 F5 3A:13 7F 47 F1 B8 28 04 32 :  29   |テVtテVt時:  G円ク( 2|
C460 :  17 64 C9 3C 32 17 64 C9:21 60 65 22 21 64 3A 0D :  CA   | d/<2 d/!'e"!d: |
C470 :  64 32 20 64 21 FF 5F 22:07 64 2A 07 64 23 23 23 :  24   |d2 d!  _" d* d###|
C480 :  23 22 07 64 7E 2A 21 64:77 23 22 21 64 3A 20 64 :  DC   |#" d~*!dw#"!d: d|
C490 :  3D 32 20 64 FE 00 20 E2:3A 0D 64 FE 01 20 09 21 :  E7   |=2 d   キ: d     !|
C4A0 :  60 65 7E 32 17 64 18 3E:3A 0D 64 32 20 64 3E 00 :  E5   |'e~2 d >: d2 d> |
C4B0 :  32 19 64 3A 19 64 47 21:60 65 22 21 64 7E B8 28 :  98   |2 d: dG!'e"!d~ク(|
C4C0 :  1F 23 E5 3A 20 64 3D 32:20 64 FE 00 28 03 E1 18 :  FA   | #◣: d=2 d  ( ト |
C4D0 :  E9 3A 19 64 3C 32 19 64:3A 0D 64 32 20 64 18 D3 :  D7   |♥: d<2 d: d2 d モ|
C4E0 :  78 32 17 64 18 00 3A 30:62 FE 03 CA AD 75 3A 00 :  30   |x2 d  :0b  ハュu: |
C4F0 :  64 4F 3A 02 64 57 3E 42:CD A4 02 CD 0E 04 E7 3E :  A1   |dO: dW>Bヘ、 ヘ  ◤>|
-----------------------------------------------------------------------------------
 SUM :  46 06 BA 35 E5 6D A0 9D E4 4A 53 57 50 01 47 CD    07

ADRS    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [0123456789ABCDEF]
-----------------------------------------------------------------------------------
C500 :  00 E7 3E 00 E7 3E 00 E7:3E 06 E7 3E 01 E7 3E 01 :  C1   | ◤> ◤> ◤> ◤> ◤> |
C510 :  E7 3E FF E7 01 00 20 21:00 40 FB 76 DB FA E6 20 :  D9   |◤> ◤     ! @ v□ ◥ |
C520 :  28 09 DB FB 77 23 0B 79:B0 20 EF DB F8 FB 76 F3 :  1B   |( □ w# y- ＼□  v月|
C530 :  CD 63 02 21 03 60 7E CD:2F 73 21 00 40 09 23 23 :  53   |ヘc ! '~ヘ/s! @ ##|
C540 :  7E FE 00 28 07 FE FF 28:03 23 18 F4 2B 7E FE 4E :  F7   |~  (    ( # 日+~ N|
C550 :  28 3E FE 9C 28 48 FE 39:28 44 FE 72 28 40 FE E4 :  CD   |(>  (H 9(D r(@ ◢|
C560 :  28 3C FE C9 28 38 FE 93:28 34 FE 27 28 30 FE 21 :  14   |(< /(8  (4 '(0 !|
C570 :  28 2C FE 42 28 28 FE 84:28 24 FE 09 28 20 FE 12 :  11   |(, B((  ($  (   |
C580 :  28 1C FE 24 28 18 FE 48:28 14 FE 90 28 10 18 BC :  C2   |(  $(  H(    (  シ|
C590 :  2B 7E FE 4E 20 B6 2B 7E:FE 4E 20 B0 23 23 47 0E :  2B   |+~ N カ+~ N -##G |
C5A0 :  00 7E 2B 0C B8 28 FA 79:32 18 64 18 05 3E 1B 32 :  5E   | ~+ ク( y2 d   > 2|
C5B0 :  18 64 3E FF 32 16 64 3E:46 32 24 64 3E 01 32 25 :  39   | d> 2 d>F2$d> 2%|
C5C0 :  64 21 00 40 22 26 64 CD:D0 76 21 60 62 22 11 64 :  FE   |d! @"&dヘミv!'b" d|
C5D0 :  E5 21 30 62 22 13 64 21:00 60 22 07 64 E1 7E FE :  9C   |◣!0b" d! `" dト~ |
C5E0 :  FF 28 29 3A 25 64 47 3A:0D 64 B8 CA E5 76 04 78 :  5E   | ():%dG: dクハ◣v x|
C5F0 :  32 25 64 23 22 11 64 E5:2A 13 64 23 22 13 64 2A :  E1   |2%d#" d◣* d#" d*|
-----------------------------------------------------------------------------------
 SUM :  B7 40 36 4E 9E 21 9C 50 3D 91 09 35 12 F1 58 C1    4E
```

```
ADRS    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM  [0123456789ABCDEF]
-----------------------------------------------------------------------------------
C600 : 07 64 23 23 23 23 22 07:64 E1 18 D2 E5 2A 13 64 : D5   | d####" dト メ◣* d|
C610 : 7E FE 00 28 06 FE 01 28:02 18 12 3A 25 64 47 3A : 41   |~  (   (    :%dG:|
C620 : 0D 64 B8 CA E5 76 04 78:32 25 64 18 CB FE 02 28 : 90   | dクハ◣v x2%d ヒ  (|
C630 : 06 FE 03 28 02 18 12 3A:25 64 47 3A 0D 64 B8 CA : 92   |  (    :%dG: dクハ|
C640 : E5 76 04 78 32 25 64 18:AF FE 04 20 05 3E 4C 32 : 3C   |◣v x2%d ッ    >L2|
C650 : 24 64 E1 22 11 64 CD 80:76 3E 46 32 24 64 2A 11 : 3C   |$dド" dへ v>F2$d* |
C660 : 64 23 3A 25 64 47 3A 0D:64 B8 CA E5 76 3A 25 64 : DC   |d#:%dG: dクハ◣v:%d|
C670 : 3C 32 25 64 E5 2A 13 64:23 22 13 64 E1 C3 DE 75 : 30   |<2%d◣* d#" dドテ"u|
C680 : 3A 00 64 4F 3A 02 64 57:3A 24 64 CD A4 02 CD 0E : F4   |: dO: dW:$dヘ、 ヘ |
C690 : 04 E7 2A 07 64 7E E7 23:7E E7 23 7E E7 23 7E E7 : 7D   | ◤* d~◤#~◤#~◤#~◤|
C6A0 : CD 2F 73 23 22 07 64 3E:01 E7 3E 01 E7 3E FF E7 : 8F   |ヘ/s#" d> ◤> ◤> ◤|
C6B0 : 2A 26 64 FB 76 DB FA E6:20 28 09 DB FB 77 23 0B : AC   |*&d v□ ◥ ( □ w# |
C6C0 : 79 B0 20 EF 22 26 64 DB:F8 FB 76 F3 CD 63 02 C9 : 16   |y- \"&d□  v月ヘc ノ|
C6D0 : 3A 00 64 4F 3A 02 64 57:3E 0A CD A4 02 CD 0E 04 : 7E   |: dO: dW> ヘ、 ヘ  |
C6E0 : E7 CD 63 02 C9 3E 00 32:2B 64 3E 00 32 2C 64 21 : 02   |◤ヘc ノ> 2+d> 2,d!|
C6F0 : 30 62 22 13 64 21 00 60:22 07 64 21 00 61 22 28 : 05   |0b" d! `" d! a"(|
-----------------------------------------------------------------------------------
 SUM : 40 0E 90 27 5B 92 28 4C C5 22 AF D8 D0 26 90 A9    03

ADRS    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM  [0123456789ABCDEF]
-----------------------------------------------------------------------------------
C700 : 64 3A 0D 64 32 2A 64 2A:13 64 7E 23 22 13 64 FE : A8   |d: d2*d* d~#" d |
C710 : 05 28 55 2A 07 64 7E 23:22 07 64 2A 28 64 77 23 : 95   | (U* d~#" d*(dw#|
C720 : 22 28 64 2A 07 64 7E 23:22 07 64 2A 28 64 77 23 : C1   |"(d* d~#" d*(dw#|
C730 : 22 28 64 2A 07 64 7E 23:22 07 64 2A 28 64 77 23 : C1   |"(d* d~#" d*(dw#|
C740 : 22 28 64 2A 07 64 7E 23:22 07 64 2A 28 64 77 23 : C1   |"(d* d~#" d*(dw#|
C750 : 22 28 64 3A 2B 64 3C 32:2B 64 3A 2A 64 FE 01 CA : 05   |"(d:+d<2+d:*d  ハ|
C760 : E6 78 3D 32 2A 64 18 9F:2A 07 64 23 23 23 7E 47 : D5   |◥x=2*d  * d###~G|
C770 : 3A 17 64 B8 CA 13 77 3A:17 64 FE 01 28 12 FE 02 : AF   |: dクハ w: d  (  |
C780 : 28 45 FE 03 28 6C FE 04:CA 11 78 FE 00 CA 24 78 : BB   |(E  (l  ハ x  ハ$x|
C790 : 2A 07 64 23 23 23 7E FE:02 28 0C FE 03 28 10 FE : E7   |* d###~  (   (  |
C7A0 : 04 28 14 FE 05 28 18 3E:01 32 2C 64 C3 77 78 3E : 74   | (   ( > 2,dテwx>|
C7B0 : 03 32 2C 64 C3 77 78 3E:06 32 2C 64 C3 77 78 3E : 6D   | 2,dテwx> 2,dテwx>|
C7C0 : 0B 32 2C 64 C3 77 78 2A:07 64 23 23 23 7E FE 03 : FC   | 2,dテwx* d###~  |
C7D0 : 28 08 FE 04 28 0C FE 05:28 10 3E 01 32 2C 64 C3 : 65   |(   (   ( > 2,dテ|
C7E0 : 77 78 3E 03 32 2C 64 C3:77 78 3E 06 32 2C 64 C3 : 6D   |wx> 2,dテwx> 2,dテ|
C7F0 : 77 78 2A 07 64 23 23 23:7E FE 04 28 04 FE 05 28 : C4   |wx* d###~  (   (|
-----------------------------------------------------------------------------------
 SUM : 8B 61 C7 2A 01 95 30 54 FE D6 29 2F 85 8A AC 40    1E

ADRS    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM  [0123456789ABCDEF]
-----------------------------------------------------------------------------------
C800 : 08 3E 01 32 2C 64 C3 77:78 3E 03 32 2C 64 C3 77 : F8   | > 2,dテwx> 2,dテw|
C810 : 78 2A 07 64 23 23 23 7E:FE 05 28 00 3E 01 32 2C : BC   |x* d###~  ( > 2,|
C820 : 64 C3 77 78 2A 07 64 23:23 23 7E FE 01 28 10 FE : C7   |dテwx* d###~  (  |
C830 : 02 28 1C FE 03 28 20 FE:04 28 24 FE 05 28 28 3A : 6A   | (   (   ($  ((:|
C840 : 18 64 FE 70 D2 6F 78 3E:01 32 2C 64 C3 77 78 3E : 94   | d pメox> 2,dテwx>|
C850 : 02 32 2C 64 C3 77 78 3E:05 32 2C 64 C3 77 78 3E : 6B   | 2,dテwx> 2,dテwx>|
C860 : 09 32 2C 64 C3 77 78 3E:13 32 2C 64 C3 77 78 3E : 80   | 2,dテwx> 2,dテwx>|
C870 : 00 32 2C 64 C3 13 77 2A:07 64 ED 5B 28 64 7E 12 : 08   | 2,dテ w* do[(d~ |
C880 : 23 13 22 07 64 ED 53 28:64 2A 07 64 ED 5B 28 64 : F8   |# " doS(d* do[(d|
C890 : 7E 12 23 13 22 07 64 ED:53 28 64 2A 07 64 ED 5B : FC   |~ # " doS(d* do[|
C8A0 : 28 64 7E 12 23 13 22 07:64 ED 53 28 64 2A 07 64 : 40   |(d~ # " doS(d* d|
C8B0 : ED 5B 28 64 7E 12 23 13:22 07 64 ED 53 28 64 3A : 2D   |o[(d~ # " doS(d:|
C8C0 : 2B 64 3C 32 2B 64 3A 2C:64 47 2A 28 64 3E FF 77 : 07   |+d<2+d:,dG*(d> w|
C8D0 : 23 77 23 77 23 77 23 22:28 64 3A 2B 64 3C 32 2B : 01   |#w#w#w#"(d:+d<2+|
C8E0 : 64 10 E7 C3 5A 77 3A 01:64 4F 3A 02 64 57 21 00 : F5   |d ◤テZw: dO: dW! |
C8F0 : 61 3E 03 32 2D 64 3E 4D:CD A4 02 CD 0E 04 E7 3A : 63   |a> 2-d>Mヘ、 ヘ  ◤:|
-----------------------------------------------------------------------------------
 SUM : D2 5A 51 D6 93 F5 1A C5 B7 6C 00 7A C6 64 CC E0    2D
```

```
ADRS    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [0123456789ABCDEF]
--------------------------------------------------------------------------------------
C900 : 17 64 E7 3A 2B 64 E7 3A:18 64 E7 3A 16 64 E7 3A : 84   | d◤:+d◤: d◤: d◤:|
C910 : 2B 64 47 FB 76 DB FA E6:20 28 2A 7E D3 FB 23 FB : DE   |+dG v□ ◥ (*~モ # |
C920 : 76 DB FA E6 20 28 1E 7E:D3 FB 23 FB 76 DB FA E6 : 32   |v□ ◥ ( ~モ # v□ ◥|
C930 : 20 28 12 7E D3 FB 23 FB:76 DB FA E6 20 28 06 7E : C1   | ( ~モ # v□ ◥ ( ~|
C940 : D3 FB 23 10 CE FB 76 F3:3A 01 64 4F 3A 02 64 57 : 18   |モ # ホ v月: dO: dW|
C950 : CD 63 02 30 24 3A 0E 7F:E6 02 20 0B 3A 2D 64 3D : 68   |^c 0$:   ◥   :-d=|
C960 : 32 2D 64 20 91 18 0A DB:F8 3E 77 32 2E 64 C3 8A : 2F   |2-d     □ >w2.dテ |
C970 : 7A 3E 64 32 2E 64 C3 8A:7A 00 3E 45 32 24 64 3E : 22   |z>d2.dテ z >E2$d>|
C980 : 01 32 2F 64 21 00 40 22:26 64 CD 75 7A 21 60 62 : 72   | 2/d! @"&d^uz!`b|
C990 : 22 11 64 21 30 62 22 13:64 21 00 60 22 07 64 2A : 1B   |" d!0b" d! `" d*|
C9A0 : 11 64 7E FE FF 28 2A 3A:2F 64 47 3A 0D 64 B8 CA : 83   | d~  (*:/dG: dクハ|
C9B0 : 8A 7A 04 78 32 2F 64 2A:11 64 23 22 11 64 2A 13 : DB   | z x2/d* d#" d* |
C9C0 : 64 23 22 13 64 2A 07 64:23 23 23 23 22 07 64 18 : E6   |d#" d* d####" d |
C9D0 : CE 2A 13 64 7E FE 00 28:06 FE 01 28 02 18 12 3A : A6   |ホ* d~  (   (   :|
C9E0 : 2F 64 47 3A 0D 64 B8 CA:8A 7A 04 78 32 2F 64 18 : 64   |/dG: dクハ z x2/d |
C9F0 : C6 FE 04 20 05 3E 49 32:24 64 CD 25 7A 3E 45 32 : 4F   |ニ    >I2$d^%z>E2|
--------------------------------------------------------------------------------------
 SUM : 09 64 BC F7 BB 96 6B 91 B4 EF 93 83 DD 95 BE FA   50

ADRS    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [0123456789ABCDEF]
--------------------------------------------------------------------------------------
CA00 : 24 64 3A 2F 64 47 3A 0D:64 B8 CA 8A 7A 3A 2F 64 : 9A   |$d:/dG: dクハ z:/d|
CA10 : 3C 32 2F 64 2A 13 64 23:22 13 64 2A 11 64 23 22 : 42   |<2/d* d#" d* d#"|
CA20 : 11 64 C3 9F 79 3A 01 64:4F 3A 02 64 57 3A 24 64 : F7   | dテ y: dO: dW:$d|
CA30 : CD A4 02 CD 0E 04 E7 2A:07 64 7E E7 23 7E E7 23 : DE   |^、 ^  ◤* d~◤#~◤#|
CA40 : 7E E7 23 7E E7 CD 2F 73:23 22 07 64 3E 01 E7 3E : 70   |~◤#~◤^/s#" d> ◤>|
CA50 : 01 E7 3E FF E7 2A 26 64:FB 76 DB FA E6 20 28 09 : 3D   | ◤> ◤*&d v□ ◥ ( |
CA60 : 7E D3 FB 23 0B 79 B0 20:EF 22 26 64 DB F8 FB 76 : A2   |~モ # y- \"&d□  v|
CA70 : F3 CD 63 02 C9 3A 01 64:4F 3A 02 64 57 3E 0A CD : E8   |月^c ノ: dO: dW> ^|
CA80 : A4 02 CD 0E 04 E7 CD 63:02 C9 3A 2E 64 DF 3A 17 : 63   |、 ^  ◤^c ノ:.d°: |
CA90 : 64 DF 3A 16 64 DF 3A 18:64 DF 3A 0D 64 DF 3E 00 : 33   |d°: d°: d°: d°> |
CAA0 : 32 2E 64 C3 C1 00 3A 01:64 4F 3A 02 64 57 3E 4D : B8   |2.dテチ : dO: dW>M|
CAB0 : CD A4 02 CD 0E 04 E7 3E:06 E7 3E 01 E7 3E FF E7 : AE   |^、 ^  ◤> ◤> ◤> ◤|
CAC0 : 3E FF E7 FB 76 DB FA E6:20 28 28 3E 00 D3 FB FB : C7   |> ◤ v□ ◥ ((> モ  |
CAD0 : 76 DB FA E6 20 28 1C 3E:00 D3 FB FB 76 DB FA E6 : CD   |v□ ◥ ( > モ  v□ ◥|
CAE0 : 20 28 10 3E 01 D3 FB FB:76 DB FA E6 20 28 04 3E : 1B   | ( > モ  v□ ◥ ( >|
CAF0 : 01 D3 FB FB 76 F3 CD 63:02 3E 55 32 2E 64 C3 8A : 09   | モ  v月^c >U2.dテ |
--------------------------------------------------------------------------------------
 SUM : 0A 94 46 6F FB D5 92 55 A0 4F 16 B4 32 3A E2 8B   9C

ADRS    +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [0123456789ABCDEF]
--------------------------------------------------------------------------------------
CB00 : 7A DB FA 2F E6 C0 FD 23:20 F7 DB FB C9 DB FA 2F : FE   |z□ /◥ヲ # 秒□ /□ /|
CB10 : E6 C0 23 20 F8 DB FB C9:00 FF 00 FF 00 FF 00 FF : 7C   |◥ヲ#  □ ノ        |
CB20 : 00 FF 00 FF 00 FF 00 FF:00 FF 00 FF 00 FF 00 FF : F8   |                |
CB30 : 00 FF 00 FF 00 FF 00 FF:00 FF 00 FF 00 FF 00 FF : F8   |                |
CB40 : 00 FF 00 FF 00 FF 00 FF:00 FF 00 FF 00 FF 00 FF : F8   |                |
CB50 : 00 FF 00 FF 00 FF 00 FF:00 FF 00 FF 00 FF 00 FF : F8   |                |
CB60 : 00 FF 00 FF 00 FF 00 FF:00 FF 00 FF 00 FF 00 FF : F8   |                |
CB70 : 00 FF 00 FF 00 FF 00 FF:00 FF 00 FF 00 FF 00 FF : F8   |                |
CB80 : FF 00 FF 00 FF 00 FF 00:FF 00 FF 00 FF 00 FF 00 : F8   |                |
CB90 : FF 00 FF 00 FF 00 FF 00:FF 00 FF 00 FF 00 FF 00 : F8   |                |
CBA0 : FF 00 FF 00 FF 00 FF 00:FF 00 FF 00 FF 00 FF 00 : F8   |                |
CBB0 : FF 00 FF 00 FF 00 FF 00:FF 00 FF 00 FF 00 FF 00 : F8   |                |
CBC0 : FF 00 FF 00 FF 00 FF 00:FF 00 FF 00 FF 00 FF 00 : F8   |                |
CBD0 : FF 00 FF 00 FF 00 FF 00:FF 00 FF 00 FF 00 FF 00 : F8   |                |
CBE0 : FF 00 FF 00 FF 00 FF 00:FF 00 FF 00 FF 00 FF 00 : F8   |                |
CBF0 : FF 00 FF 00 FF 00 FF 00:FF 00 FF 00 FF 00 FF 3E : 36   |               >|
--------------------------------------------------------------------------------------
 SUM : 58 95 15 49 D6 95 F0 E6 18 F0 D3 F4 C1 D4 F2 66   48
```

```
ADRS   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  SUM  [0123456789ABCDEF]
--------------------------------------------------------------------------------
CC00 : 91 D3 FF 3E 0C CD 2D CC:3E 70 CD 33 CC 3E 00 CD : F8  | モ > ヘ-フ>pヘ3フ> ヘ|
CC10 : 33 CC 3E 0B CD 33 CC 3E:19 CD 33 CC 21 00 C0 01 : 19  |3フ> ヘ3フ> ヘ3フ! タ |
CC20 : 18 0B 7E CD 33 CC 23 0B:79 B0 20 F6 C9 F5 3E 0F : E5  |   ~ヘ3フ# y- 分ノ時> |
CC30 : D3 FF 3E F5 DB FE E6 02:28 FA 3E 0E D3 FF F1 F5 : EC  |モ >時ロ ◥ ( > モ 円時|
CC40 : D3 FD 3E 09 D3 FF DB FE:E6 04 28 FA 3E 08 D3 FF : E6  |モ > モ ロ ◥ ( > モ |
CC50 : DB FE E6 04 20 FA F1 C9:00 FF 00 FF 00 FF 00 FF : 93  |ロ ◥   円ノ         |
CC60 : 00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00 : 00  |                  |
CC70 : 00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00 : 00  |                  |
CC80 : FF FF FF F7 FF FF FF FF:FF FF FF FF FF FF FF FF : E8  |    秒             |
CC90 : FF FF FF FF FF FF FF FF:FF FF FF FF FF FF FF FF : F0  |                  |
CCA0 : FF FF FF FF FF FF FF FF:FF FF FF FF FF FF FF FF : F0  |                  |
CCB0 : FF FF FF FF FF FF FF FF:FF FF FF FF FF FF FF FF : F0  |                  |
CCC0 : 08 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00 : 08  |                  |
CCD0 : 00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00 : 00  |                  |
CCE0 : 00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00 : 00  |                  |
```

※マシン語の打ち込みはC000H〜CC5FH

●使いかた

まず、READ する DRIVE ナンバーを入力します。(マスター・ディスケットが入っている DRIVE ナンバー)次に WRITE する DRIVE ナンバーを入力してください。(コピーする側の DRIVE ナンバー)

それぞれ1〜4の数字を入れてください。このプログラムでは1⇨2、2⇨1だけでなく1⇨4、2⇨3、3⇨4などもできます。ただしシングル・ドライブには対応していません。

次にバックアップする START TRACK ナンバーとEND TRACK ナンバーを入力してください。最後にディスケットのフォーマットをする時のパラメータを表示するかどうかで〝Y〞(Yes)、〝N〞(No)またはリターン・キーを押します。

パラメータを表示しなければ、リターン・キー5回でバックアップ開始です。この場合は DRIVE 設定：1⇨2、START TRACK：0、END TRACK：79、パラメータ表示：No の設定となります。

●プログラムの説明

BASIC 部ではイニシャライズ、パラメータ入力、DISK側プログラム・スタート、パラメータ転送、パラメータ受け取り、トラック・ナンバー、FORMAT パラメータ表示などを行なっています。

マシン語部では、ここで詳しい解説をしてみましょう。

ところで、マシン語プログラムは、主に次にあげる4つの動作に別れています。

アナライズ、リード・データ、フォーマット、ライト・データです。(その他には DISK プログラム転送、TRACK イレースがある)

①アナライズ

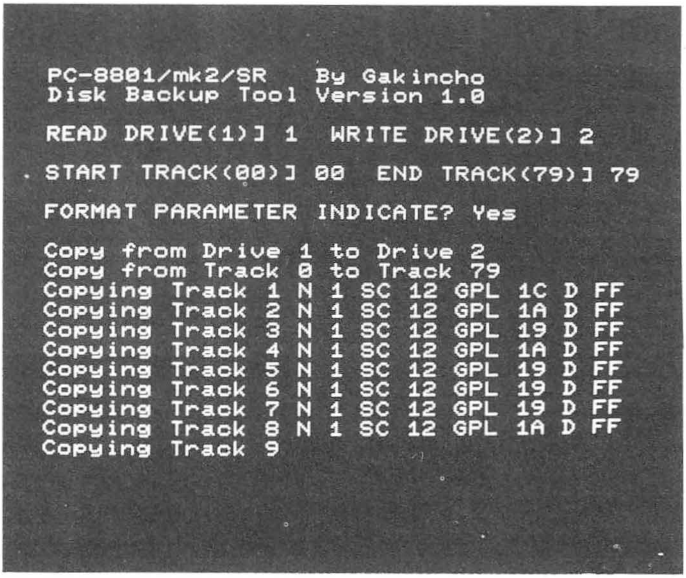
最初に READ/WRITE する DRIVE のヘッドをシークさせ時間を計りながら ID を読み込みます。次にIDの繰り返しと時間の両方でセクタ数を求めます。

次にIDの SECTOR 長をチェック(0〜6 OK、7〜255NG)します。その後はセクタ数分だけ各1セクターをリードし RESULT STATUS をバッファにストアします。その

RESULT STATUSからデータCRCエラーの起こっているセクタを捜しリードします。

リードしたデータをREAD/WRITE BUFFERに入っているバッファにうめてあるデータを読みデータの数を教えてそれをセクタ長としてカウントし、GAP長も続いてカウントします。

データCRCエラーが起こっていない場合には、IDのセクタ長からFORMAT時のセクタ長を求めます。GAP長はREAD DIAGNOSTICを実行し、最初のセクタのGAP3をカウントしGAP長とします。DATAはFFHとなります。



プログラムをスタートさせると……。

②リード・データ

エラーのない正常なセクタをリードしREAD/WRITE BUFFERにストアします。

③フォーマット

最初にIDを調べてダミー・セクタを入れるかどうかをチェックし、ダミーを入れる必要があるのならダミー・セクタを混入します。その後、IDをWRITE ID BUFFERに移し、ダミー数+セクタ数分だけWRITE ID BUFFERからIDをFDCに送ってフォーマットします。

④ライト・データ

エラーのない正常なセクタをWRITEします。

⑤ TRACKイレース

IDがないとFDCはアンフォーマット判断とするのでIDを消すためにセクタ長6でフォーマットしイレースします。

⑥ DISKプログラム転送

DISK側のプログラム(C000H〜)をDISKドライブの7000Hに転送します。

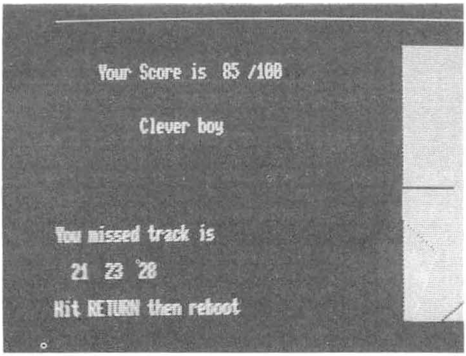
なお、バッファ・アドレスは次の通りです。

6000H〜:READ ID FBUFFER
6100H〜:WRITE ID BUFFER
6200H〜:ID TIME BUFFER
6230H〜:RESULT STATUS BUFFER
6260H〜:SECTOR LENGTH CHECK BUFFER
4000H〜:READ/WRUTE BUFFER

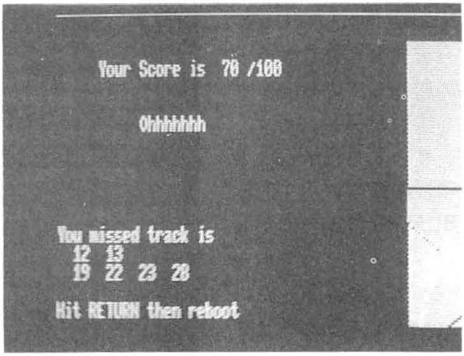
ワーク・エリアは以下の通りです。

6400H〜:READ DRIVE No.
6401H〜:WRITE DRIVE No.
6402H〜:TRACK No.
640DH〜:SECTOR数
6416H〜:DATA
6417H〜:SECTOR長
6418H〜:GAP数

以上の全てはDISK側のアドレスです。なお,本プログラムの動作を確認している機種は、PC-8801mk II,PC-8801mk II SR、PC-80S31、K-305PC、TF-20です。



〝RATS & STAR〞対〝HAND PICK〞



〝RATS & STAR〞対〝オニャンコN88〞

●最後に

このプログラムでは、インデックス・ホールから最初のセクタのTIME使用していません。このTIMEを使えば2周フォーマットやTOP SHIFTなどのプロテクトも解読できます。

さて、ここまで読んでみてどうでしたか？いろいろな専門用語がでてきますが、これらはμPD765、726ユーザー・マニュアルや秀和から出ている〝フロッピーディスク活用研究〞などを読んで理解してください。

そして、今度は自分自身で本プログラムのバージョンUPをしてみてください。

●編集部から

いかがでしたでしょう。もうプログラムは入力されましたか？さて、DISKバックアップ・ツールで気になるのは、どのくらいの強さなのかということでしょう。

最初にもありましたように、多様化しているプロテクト・テクニックに対応するためには、どうしてもオートモードよりもファイラー形成のツールが有利になってきます。しかしファイラーに負けるといっても、それなりの使いかたをすればまだまだ実用になるはずです。

ところで、問題の強さですが今をときめく〝The HAND PICK〞(B1.0)を取り出し〝RATS & STAR〞のバックアップにチャレンジし、本プログラムと比較してみることにしました。この〝RATS & STAR〞はコピー・ツールで落ちない場合には、ツールで取れなかったTRACKを表示してくれ、さらに100分のいくつの数字でツールの強さを表示します。（いかに強いプロテクトをかけてあるか、取れるなら取ってみろといわんばかりの自信のほどがうかがえます）

結果は、HAND PICKがコピーできず100分の85、本プログラムもコピーできず100分の70と出ました。この結果からも初回としてはマアマアの線だろうということになります。

今後もこのバックアップ・ツール〝オニャンコN88〞のバージョンUPをはかりたいと考えています。また、このツールでどんなソフトが落とせたかを編集部にお知らせください。





若松通商

〒101 東京都千代田区外神田1-9-9
内田ビル3階☎03-253-8521

国電秋葉原駅から総房線のガード下をくぐり、右手に秋葉原ラジオデパートを見ながら少し歩くと右に若松通商（マイコンパーツセンター）のある内田ビルが見付かります。隣には〝ネジの西川〟というお店があるのですぐ分かるはずです。

若松通商といえば、秋葉原の常連なら『アア、ラジオ会館やミツワ・ビルにあるお店だ！』と即座に答えられるほど有名なショップですが、このうちの1つが当店、マイコンパーツセンターです。

さて、店内をぐるりと見回すと目につくのがRATS & STARやMAGIC COPYなどのツールがズラリ並んでいるコーナーです。お店の話しでは「秋葉原ではウチと同じくらいツールを扱っているところは少ないでしょう」と自慢顔。

実際に本書で紹介しているツールのほとんどが置いてあるのにはビックリです。また、〝たーぼジョイスティック〟の製作等には欠かせない小物パーツも揃えてあります。

その他のコーナーには、各メーカーの最新機種が並び、我らハッカーには見逃せないショップです。また、地方の方には嬉しい通信販売も扱っていますから、直接お店に連絡すれば便宜をはかってもらえます。

ちょっと目につかない穴場といえそうなマイコン・ショップです。もし秋葉原に寄るんだったらこのお店に足を運んでみてください。マル秘情報が手に入るかも!!



ソフマップ

〒101 東京都千代田区外神田
３－５－４
末広町ハイム40↑
☎03-253-4226
☎03-496-2374

国電秋葉原駅から中央通りを上野方向に歩いて行くと左にラジオ・ファンのメッカ、カクタＸ１（シャープみたいですね）のビルがあります。その前を通りすぎてもう少し歩くとハッカーらしき少年達でいっぱいのお店に出くわします。

このマイコン・ショップがソフマップです。いまや全国のハッカー連中が、注目しているほどの魅力あるお店です。というのも我ら貧乏ハッカーには、手も足もでなかったＰＣ、ＦＭ、Ｘ１といったパソコンがお小遣いで入手できる値段で売り出されているからです。

現行商品でさえ30～40％の値引きです。それでも予算が足りない人には中古パソコンをお勧めします。店内に所狭しと置かれた中古品は５割引きの価格は当たり前というほどです。

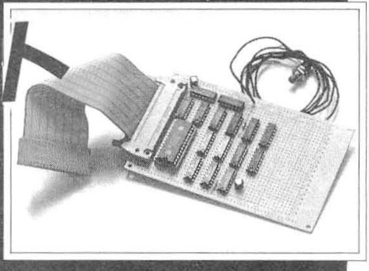
秋葉原でも行って徳するお店はそんなに多くはありませんが、このソフマップは数少ないお店のうちの１つに数えてもよいでしょう。

秋葉原ショップ情報

PC用両面倍密ドライブ(PC-80S31/相当品)

コピー・プロテクト その傾向と対策

川本幸治



この本を読まれる皆さんは、きっとコピープロテクトやバックアップに興味があると思います。しかし、どうやってバックアップしていますか？

オート only？それじゃ今はだめでしょう。ファイラー？それだと何か他人にやってもらうような気がしませんか？コピー・マシン？う〜ん、リッチですね。

アナライザー(マニュアルモード)ほ〜正統派ですね。なんだって？使い方がよくわからない？そうですか。それじゃ宝の持ち腐れですよ。

それでは、使い方を説明しましょう。一体何がわからないのですか!?　ふんふん、なるほど。

というわけで、プロテクトの傾向と対策について説明したいと思います。

はっきり言って、プロテクトについて全部説明すると、1冊の本（どころではない）ができてしまうので、他機種や専用機によるプロテクトは省略し、ＰＣの、ＰＣによる、ＰＣのためのプロテクトのみについて、アナライザーを使用したコピーの仕方を説明したいと思います。

………… 実践編 …………

さて、ＰＣだけでできるプロテクトには、どのようなものがあるのでしょうか。

(1)ＩＤ情報を変化させる
(2)オーバー/アンダーセクタ
(3)オーバートラック
(4)オーバーインデックス
(5)セクタ長混在/CRC エラー
(6)書き継ぎのチェック
(7)単密・倍密混在/ＣＲＣエラー

ぐらいのものだと思います。(誰ですか？(7)はＰＣでは不可能だ、なんて言っている人は？実はできるのです)

このうち、(1)(2)(3)は、どんなアナライザでも簡単にできます。（説明書を読めばわかる）また、(7)は少し難しいので、次の機会にまわすことにします。

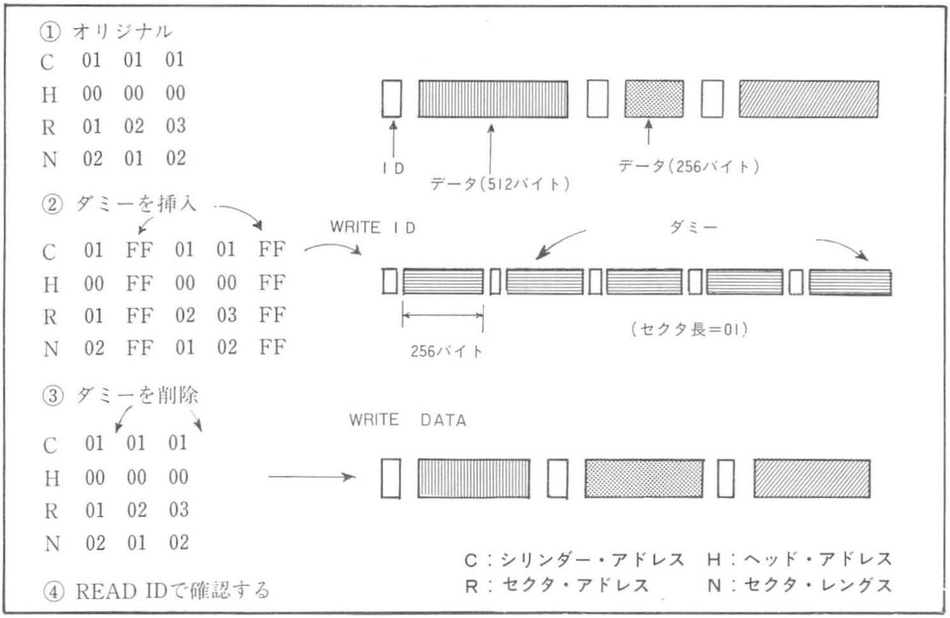
というわけで、わかりにくいと思われる(4)(5)(6)について説明しようと思います。また、全く新しい考えによるハード・アナライザー「Head Chaser」についても説明しようと思います。

……セクタ長混在……

ＰＣでは、ハード（ＦＤＣ）の制約から、フォーマット時にセクタ長混在を作ることは不可能です。では、どうすればいいのか。

答えは簡単です。ダミーを挿入すればいいのです。図１を見てください。①のようなフォーマットを作るのには、次のようにすればいいのです。

(1)まず、READ ID (SCAN ID) 及び READ DATA を行ないます。
(2)次に WRITE ID (FORMAT) を行なうわけ



〈図1〉 セクタ長混在

ですが、その前にEDIT IDなどにより、ダミーを挿入します。その後にWRITE IDを行ないます。

(3)ダミーを削除し、WRITE DATAを行ないます。

(4) SOURCEとDESTINATIONを入れ替え、READ IDで確認します。フォーマット時のダミーが消去されて、元のようなフォーマットができあがっているはずです。

一般的には、一番小さい「N」のセクタ長でフォーマットすればよいようです。GAP3や、ダミーの数をいくつにしたらいいか、ということが問題となりますが、そこは自分で何度もやってみた方がいいと思います。

とは言っても、うまくいかなかった時にどうすればいいかわかる人は少ないと思いますので、少しだけヒントを書いておきますので参考にしてください。

もし、セクタがいくつか消滅してしまったら、その前のダミーか、GAP長を増やしてください。また、後ろの方のセクタが残ったり、全部消えてしまったりしたら、それは1トラックの中に入り切らず、2周目にかかってしまったのですから、GAP長を減らしてください。

全く同じフォーマットを作れるようになるには、経験が必要です。みなさん、がんばってください。

‥‥‥‥CRCエラー‥‥‥‥

PCでは、実際のセクタ長と違うNをフォーマットのときにつけてフォーマットすると、それを読んで時にCRCエラーが起こります。(図2参照)

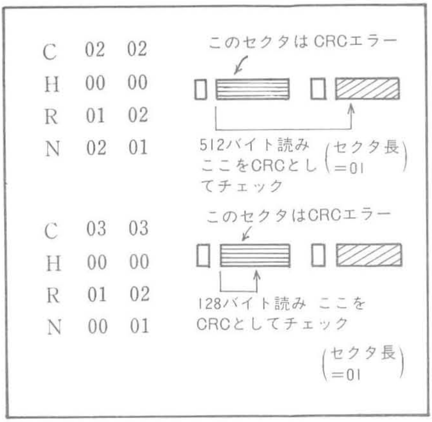
これを作るには、次のようにすればいいのです。

(1) READ ID、WRITE IDを行なう

(2)CRCエラーのセクタのIDを削除。

(3) READ DATA、WRITE DATAを行なう。

なお、フォーマットのときのデータ長の決め方ですが、そのセクタのデータを読んで表示して(READ DATA、PRINT DATA使用)、



〈図2〉 CRCエラー

(1)途中まで同じデータの時は、そこまでをデータ長とする。
(2)最後まで同じデータの時は、データ長を、「N」より1つ（2倍）大きくする。

なお、これで求めたデータ長が、他のセクタのデータ長より短い時は、ダミーを挿入(「セクタ長混在」参照）してください。

単倍のテクニックを使ってＣＲＣエラーを作る方法もありますが、それは次の機会に回したいと思います。

…書き継ぎチェック…

ディスクというものは、フォーマット後一度でも書き込みを行なうと、必ずその前後にずれができます。そのずれが有るか無いかをチェックするプロテクトです。(図3)

コピー方法は、次のとおりです。

(1) READ ID、WRITE ID を行なう。(FORMAT DATA は、ずれのないセクタのものを使う)
(2)ずれのないセクタを削除。
(3) READ DATA、WRITE DATA を行なう。

チェック法は、次のようにします。

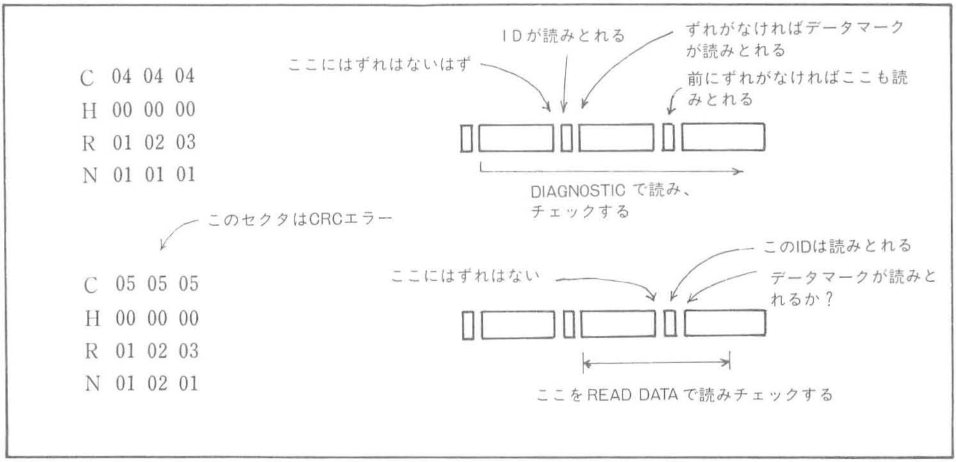
(1)N＝6で READ DIAGNOSTIC を行ない、DATA を表示してみます。

最初のセクタに相当する部分が、もし一定のデータで埋められていたら要注意です。ＧＡＰが「４Ｅ」で、次のＩＤが読みとれるようなら、そのセクタには書き込みが行なわれていないという証拠です。

また、次のセクタのデータマークが読みとれるようなら、次のセクタにも書き込みは行なわれていません。(その次以降も同じ)

(2)ＣＲＣエラーが起こるセクタの READ DATA を行ない、表示してみます。

そのセクタは一定のデータで埋められ、次



〈図3〉 書き継ぎチェック

のセクタのＩＤが読みとれると思います。もし、次のセクタのデータマークが読みとれ、そのセクタが一定のデータで埋められていれば、次のセクタに書き込みが行なわれていません。

…オーバー・インデックス…

このフォーマットには、次の２つがあります。（図６）

(1)トップシフト

(2)２周フォーマット

これらのフォーマットのかけ方は簡単です。

トップシフト

(1) READ ID、READ DATA を行なう。

(2)一番最初にダミーを１つ挿入する。

(3) WRITE ID を行なう。

(4)ダミーを消去し、WRITE DATA を行なう。

(5)ダミーが消えていることを確認する。(READ ID 使用)

ダミーが消えない場合は、GAP3 を増やしてみてください。また、他のセクタまで消えてしまったら、GAP3 をへらしてください。

２周フォーマット

(1) READ ID、READ DATA を行なう。

(2)一番最初にダミーをいくつか挿入する。

(3) WRITE ID を行なう。

(4)ダミーを消去し、WRITE DATA を行なう。

(5) READ ID で確認する。

ダミーの数ですが、最初は ID の数と同じにしてフォーマットしてみてください。そして、

```
40A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4100 FB E5 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
4110 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
4120 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
4130 4E 4E 4E 4E 4E 00 00 00 00 00 00 00 00 00 00 00
4140 00 A1 A1 A1 FE 04 00 02 01 65 AE 4E 4E 4E 4E 4E
4150 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
4160 4E 00 00 00 00 00 00 00 00 00 00 00 00 A1 A1 A1
4170 FB FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4180 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4190 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
41F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4200 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4210 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4220 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4230 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4240 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4250 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4260 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
4270 FF FB E5 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
4280 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
4290 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
42A0 4E 4E 4E 4E 4E 4E 00 00 00 00 00 00 00 00 00 00
42B0 00 00 A1 A1 A1 FE 04 00 03 01 56 9F 4E 4E 4E 4E
```

IDアドレスマーク
２セクタ目 ID
ID CRC
１セクタ目 CRC
SYNC
GAP
２セクタ目 データマーク
２セクタ目 CRC

〈図４〉 ２セクタ目に書き込んでない場合

```
40B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
40F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ← このあたりは図4と同様（2
4100 FB E5 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   セクタ目IDは読みとれる）
4110 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
4120 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
4130 4E 4E 4E 4E 4E 00 00 00 00 00 00 00 00 00 00 00
4140 00 A1 A1 A1 FE 04 00 02 01 65 AE 4E 4E 4E 4E 4E  SYNC ビットずれが起こってい
4150 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  る
4160 4E 07 FF FF FF FF FF FF FF FF FF FF FF F0 A0 A0 ← 2セクタ目データマーク
4170 A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ここもビットずれが起こって
4180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   いる
4190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
41D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ← ビットずれで「FF」が「00」に
41E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   なっている
41F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 — 2セクタ目CRC(ビットずれ)
4270 00 00 00 81 0F 27 27 27 27 27 27 27 27 27 27 27
4280 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27  GAP 3 本当は「4E」
4290 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27
42A0 27 27 27 27 27 27 00 00 00 00 00 00 00 00 00 00
42B0 00 00 50 D0 D0 FF 02 00 01 80 AB 4F A7 27 27 27
```

〈図5〉 2セクタ目に書き込んだ場合

最初の方のセクタがいくつか消えると思いますので、その数だけダミーを増やしてやり直してみてください。

しかし、オーバー・インデックスがかかっていることを見付けるのは至難の技だと思います。（相当の経験とカンを要するでしょう）

そこで、「Head Chaser」の登場とあいなります。

「いったい、それは何だ？」と思われるでしょう。実は、ハードによるアナライザなのです。

「ハードによるアナライザだって？いったいどんな原理でやっているんだ？」

あわてない、あわてない。今から説明します。

……Head Chaser……

一言で言ってしまえば、「高級トラック・カウンタ」です。

ソフトの起動のときにＦＤＣに送られたコマンド、ＦＤＣから送られたステータスをすべて表示してしまうものです。

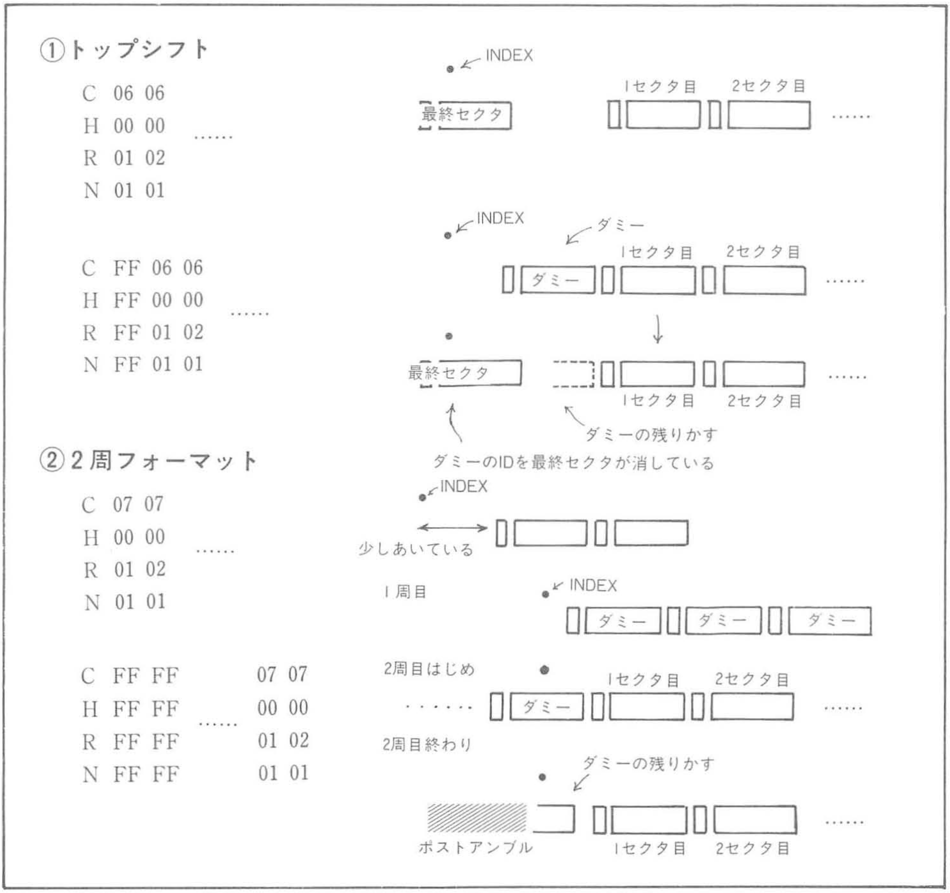
原理は、ＦＤＣとＣＰＵのやりとりをすべて横から見ていて、メモリにたくわえてしまうというものです。

「いったいどうやって使うんだ？出てきたデータの表示方法は？それをどうやって読むんだ？」本当にあなたは気の短い人ですね。順番に説明しますよ。

使い方は大変簡単です。

(1)リセット・ボタンを押して、解析したソフトを起動させます。

(2)起動したら（あるいは停止したら）専用のソフトに入れ替えて、基板のモード・スイッ



〈図6〉 オーバー・インデックス

チを押しながらリセット・ボタンを押します。
(3)データをセーブ、またはプリントします。

たったこれだけのことです。もちろん、データの表示も専用のソフトでやっています。

さて、読み方ですが、要注意は「READ DIAGNOSTIC」と「READ ID」です。この2つに注意すれば、大体のプロテクトはわかります。オーバー・インデックスのチェック法ですが、

(1)別のトラックで、READ DIAGNOSTIC
(2)そのトラックにSEEK
(3)そのトラックで、READ　ID

と言うのが、一般的なパターンです。(もっとも、(1)のかわりにREAD DATAを使う方法もあります)

従って、こういうところを見付けたら、まずオーバー・インデックスがかかっていると見ていいでしょう。

えっ?何ですかア?もっと確実に見付けたいって?そんな時は、まず普通にコピーしてみて両方を比較してみればよろしい。

プリンタの無い人のために、コンペアの機能も用意されていますから、それを使用するといいでしょう。

(1)オリジナルとコピー両方のデータを取ります。

「Head Chaser」での解析例

```
1 SPECIFY
DA 19

2 RECALIBLATE
DRIVE= 0

3 SENSE INTERRUPT STATUS
ST0=20 CYLINDER=0

4 SEEK
HEAD= 0 DRIVE= 0
NEW CYLINDER NUMBER=A

5 SENSE INTERRUPT STATUS
ST0=20 CYLINDER=A

6 RECALIBLATE
DRIVE= 0

7 SENSE INTERRUPT STATUS
ST0=20 CYLINDER=0

8 RECALIBLATE
DRIVE= 1
```

オーバーインデックスのチェック例

```
READ DIAGNOSTIC
MFM
HEAD= 1 DRIVE= 0
C=D H=1 R=1 N=2 EOT=10 GPL=33 DTL=FF
ST0=4 ST1=24 ST2=20
C=D H=1 R=2 N=2

SEEK
HEAD= 0 DRIVE= 0
NEW CYLINDER NUMBER=13

SENSE INTERRUPT STATUS
ST0=20 CYLINDER=13

READ ID

HEAD= 0 DRIVE= 0
ST0=0 ST1=0 ST2=0
C=13 H=0 R=F N=1

READ ID

HEAD= 0 DRIVE= 0
ST0=0 ST1=0 ST2=0
C=13 H=0 R=10 N=1

READ ID
```

(2)オリジナルのデータをロードします。
(3)コンペアモードを指定し、ファイルネームにコピーのファイルネームを指定します。
(4)一致すれば「ALL EQUAL!」と表示し、一致しなければデータが画面に出力され、途中で「UNEQUAL!!」と表示して止まりますので、その直前のデータを見てください。

2つを較べてみれば、どこが違うかわかるので、それを参考にコピーしてください。(なおチェック・ルーチンが甘ければ、IDをローテートするだけでとれてしまいます。)

もちろん、調べに行かないトラックは正直にコピーする必要はありません。(但し、もっと後で調べに行く場合もあります。)

これでデータを取れば、プロテクト解析も早くなることでしょう。

また、「アインシュタイン」等との併用もいいですよ。どこのトラックが異常か分かれば、割と簡単にコピーすることができますから。

一応、回路図とプログラムリストを載せておきますが、ハードに自信のない人は製作しない方がよいでしょう。(本体を壊しても知りませんよ!!)

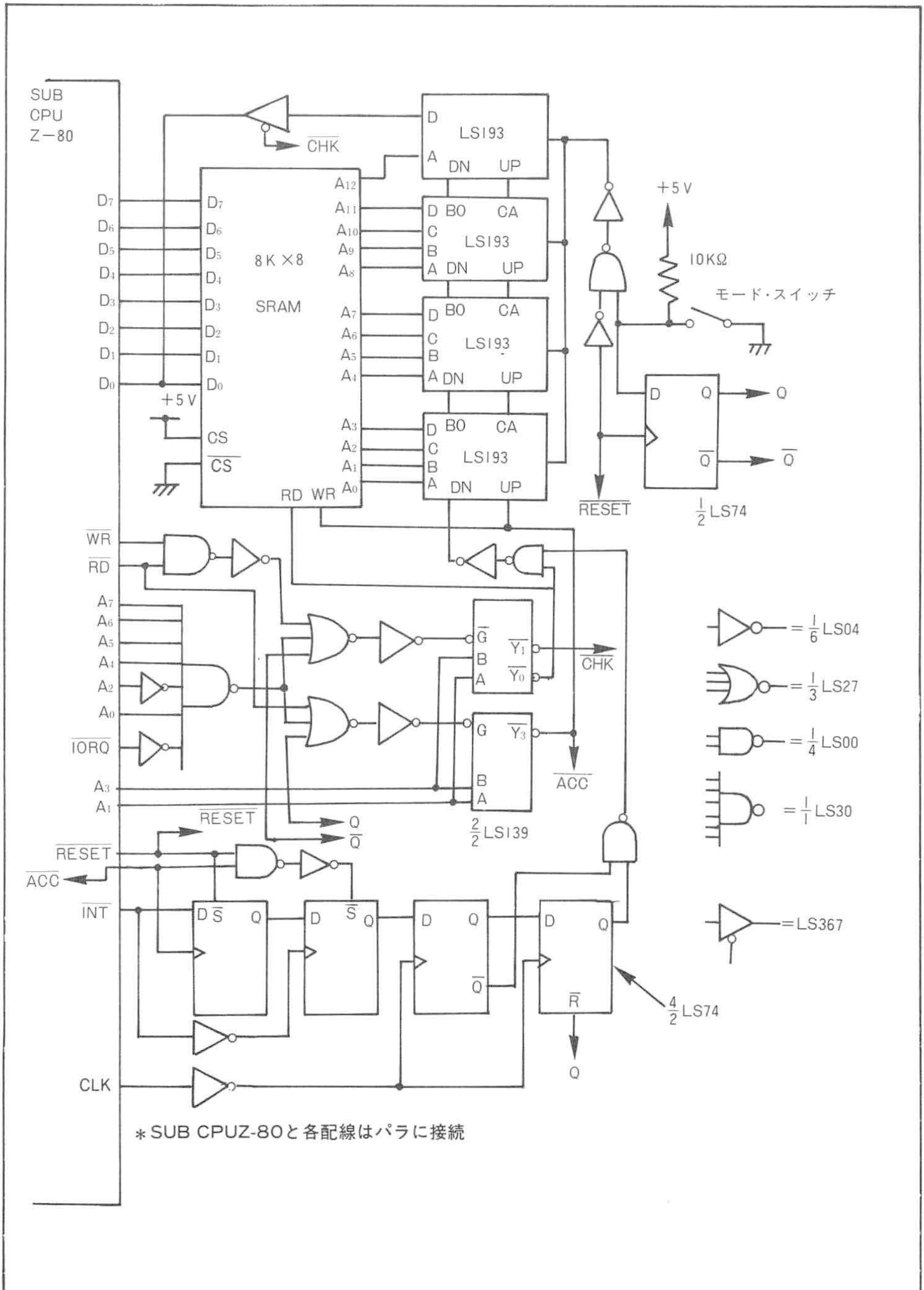
………終わりに………

みなさんも、アナライザをどんどん使い、経験を積んでカンを養ってください。きっと、あなたもロックキラーの仲間入りができるでしょう。

もし、みなさんのご希望があれば、次回は(7)の単倍フォーマットについて説明したいと思います。(僕の知る限りでは、今まで誰も考えついた人はいません)

また、「Head Chaser」についてですが、ただいま商品化の交渉を進めていますので、しばらくお待ちください。(あるいは、これが出るころには出せるかも知れない)

「いいアナライザが欲しい」だって?ご要望が多ければ次回まで何とかしましょう。



〈図７〉「Head Chaser」の回路図

〈Head Chaserのプログラム・リスト〉

```
1 REMOVE:CLEAR 300,&HABFF:DEFINT A-Z
2 WIDTH 80,25
30 AD=&HCC50:RESTORE 50080:GOSUB 50010:DEF USR0=&HCC50:DEF USR1=&HCC56
40 AD=&HCC00:RESTORE 50130:GOSUB 50010:DEF USR2=&HCC00
50 E=USR0(&HC):RESTORE 50180:GOSUB 50050
60 E=USR0(&HD):E=USR1(&H50):E=USR1(&H0)
70 EA=&HCBFF:SA=USR2(EA)
80 '
90 PRINT
95 PRINT"1  ...LOAD DATA
100 PRINT"2  ...PRINT DATA
105 PRINT"3  ...FILES
106 PRINT"4  ...COMPARE
107 PRINT"5  ...SAVE DATA
108 PRINT"?";
110 CP=0:E=VAL(INPUT$(1)):IF E>0 AND E<=5 THEN PRINT E:ON E GOTO 500,1000,700,80
0,200
120 GOTO 110
199 'SAVE DATA
200 IF SA=EA THEN PRINT"NO DATA !!":BEEP:GOTO 90
205 REMOVE:MOUNT 1:PRINT"Data Save"
210 INPUT"File Name";F$
220 OPEN F$ FOR OUTPUT AS #1
240 ST=SA:EN=EA
250 PRINT#1,MKI$(EA-SA);
260 A$="A":AD=0:AA=VARPTR(A$):AB=VARPTR(AD):POKEAA,255
270 AD=ST:IF ST>EN-255 THEN 310 ELSE FOR AD=ST TO EN-255 STEP 255
280 POKE AA+1,PEEK(AB):POKE AA+2,PEEK(AB+1)
290 PRINT#1,A$;
300 NEXT AD
310 LO=EN-AD+1:POKE AA,LO
320 POKE AA+1,PEEK(AB):POKE AA+2,PEEK(AB+1)
330 PRINT#1,A$
340 CLOSE #1:A$="":REMOVE:GOTO 90
499 'LOAD DATA
500 REMOVE:MOUNT 1
510 INPUT"File Name";F$
520 OPEN F$ FOR INPUT AS #1
530 ST=&HAC00:EN=ST+CVI(INPUT$(2,1)):SA=ST:EA=EN
540 A$="A":AD=0:AA=VARPTR(A$):AB=VARPTR(AD):POKEAA,255
550 AD=ST:IF ST>EN-255 THEN 590 ELSE FOR AD=ST TO EN-255 STEP 255
560 POKE AA+1,PEEK(AB):POKE AA+2,PEEK(AB+1)
570 LSET A$=INPUT$(255,1)
580 NEXT AD
590 LO=EN-AD+1:POKE AA,LO
600 POKE AA+1,PEEK(AB):POKE AA+2,PEEK(AB+1)
610 LSET A$=INPUT$(LO,1)
620 CLOSE #1:A$="":REMOVE:GOTO 90
699 'FILES
700 PRINT:MOUNT 1:FILES:REMOVE:GOTO 90
799 'COMPARE
800 CP=1:INPUT"File Name";F$
810 REMOVE:MOUNT 1:OPEN F$ FOR INPUT AS #1
820 FOR CA=SA TO SA+CVI(INPUT$(2,1))
830 IF PEEK(CA)<>ASC(INPUT$(1,1)) THEN CLOSE:REMOVE:GOTO 1020
840 NEXT:CLOSE:REMOVE:PRINT:PRINT"ALL EQUAL !!":GOTO 90
1000 PRINT"Printer?";
1010 A$=INPUT$(1):IF A$="Y" OR A$="y" THEN POKE &HF1F2,&HC3 ELSE IF A$="N" OR A$
="n"THEN POKE &HF1F2,&HC9 ELSE 1010
1015 IF A$="N" OR A$="n" THEN PRINT:PRINT"PUSH SPACE KEY TO START":IF INPUT$(1)<'
>" " THEN 1015
1020 POKE &HF1F3,&H3A:POKE &HF1F4,&H47:AD=SA:L=0
1030 PRINT:L=L+1:PRINT MID$(STR$(L),2);" ";:DA=PEEK(AD):AD=AD+1
1035 ON(DAAND31)+1 GOSUB 5000,5000,2160,2280,2260,2120,2100,2230,2240,2150,2300,
5000,2110,2340,5000,2210,5000,2180,5000,5000,5000,5000,5000,5000,5000,2190,5000,
5000,5000,2200,5000,5000
1040 IF CP=1 AND AD>CA THEN PRINT:PRINT"UNEQUAL !!!":BEEP:GOTO 90
1050 IF AD>EA THEN POKE &HF1F2,&HC9:GOTO 90 ELSE 1030
1999 'RW GROUP
2000 IF DA AND 128 THEN PRINT"MULTI TRACK ";
2010 IF DA AND 64 THEN PRINT"MFM ";
2020 IF DA AND 32 THEN PRINT"SKIP ";
2030 PRINT:DA=PEEK(AD):AD=AD+1:PRINT"HEAD=";(DA AND 4)/4;"DRIVE=";DA AND 3
2040 PRINT"C=";HEX$(PEEK(AD));" H=";HEX$(PEEK(AD+1));" R=";HEX$(PEEK(AD+2));" N=
";HEX$(PEEK(AD+3));
```

このプログラムはN-BASICで動作しています。なお、How many file? は2にセットしてください。

```
2050 PRINT" EOT=";HEX$(PEEK(AD+4));" GPL=";HEX$(PEEK(AD+5));" DTL=";HEX$(PEEK(AD
+6))
2060 PRINT"ST0=";HEX$(PEEK(AD+7));" ST1=";HEX$(PEEK(AD+8));" ST2=";HEX$(PEEK(AD+
9))
2070 PRINT"C=";HEX$(PEEK(AD+10));" H=";HEX$(PEEK(AD+11));" R=";HEX$(PEEK(AD+12))
;" N=";HEX$(PEEK(AD+13))
2080 AD=AD+14:RETURN
2099 'READ DATA
2100 PRINT"READ DATA":GOTO 2000
2109 'READ DELETED DATA
2110 PRINT"READ DELETED DATA":GOTO 2000
2119 'WRITE DATA
2120 IF(DA AND 32)<>0 THEN GOTO 5000
2130 PRINT"WRITE DATA":GOTO 2000
2139 'WRITE DELETED DATA
2140 IF(DA AND 32)<>0 THEN GOTO 5000
2150 PRINT"WRITE DELETED DATA":GOTO 2000
2159 'READ DIAGNOSTIC
2160 IF(DA AND 32)<>0 OR (DA AND 128)<>0 THEN GOTO 5000
2170 PRINT"READ DIAGNOSTIC":GOTO 2000
2179 'SCAN EQUAL
2180 PRINT"SCAN EQUAL":GOTO 2000
2189 'SCAN LOW OR EQUAL
2190 PRINT"SCAN LOW OR EQUAL":GOTO 2000
2199 'SCAN HIGH OR EQUAL
2200 PRINT"SCAN HIGH OR EQUAL":GOTO 2000
2209 'SEEK
2210 PRINT"SEEK":PRINT"HEAD="; (PEEK(AD)AND4)/4;"DRIVE=";PEEK(AD)AND3
2220 PRINT"NEW CYLINDER NUMBER=";HEX$(PEEK(AD+1)):AD=AD+2:RETURN
2229 'RECALIBLATE
2230 PRINT"RECALIBLATE":PRINT"DRIVE=";PEEK(AD)AND3:AD=AD+1:RETURN
2239 'SENSE INTERRUPT STATUS
2240 PRINT"SENSE INTERRUPT STATUS"
2250 PRINT"ST0=";HEX$(PEEK(AD));:IF (PEEK(AD)AND192)=128 THEN PRINT" *INVALID":A
D=AD+1:RETURN ELSE PRINT" CYLINDER=";HEX$(PEEK(AD+1)):AD=AD+2:RETURN
2259 'SENSE DEVICE STATUS
2260 PRINT"SENSE DEVICE STATUS"
2270 PRINT"DRIVE=";PEEK(AD)AND3;"ST3=";HEX$(PEEK(AD+1)):AD=AD+2:RETURN
2279 'SPECIFY
2280 PRINT"SPECIFY"
2290 PRINTHEX$(PEEK(AD));" ";HEX$(PEEK(AD+1)):AD=AD+2:RETURN
2299 'READ ID
2300 IF(DA AND 32)<>0 OR (DA AND 128)<>0 THEN 5000
2310 PRINT"READ ID":IF DA AND 64 THEN PRINT"MFM";
2320 PRINT:DA=PEEK(AD):PRINT"HEAD=";(DA AND 4)/4:"DRIVE=":DA AND 3
2330 AD=AD-6:GOTO 2060
2339 'WRITE ID
2340 IF(DA AND 32)<>0 OR (DA AND 128)<>0 THEN 5000
2350 PRINT"WRITE ID":IF DA AND 64 THEN PRINT"MFM";
2360 PRINT:DA=PEEK(AD):PRINT"HEAD=";(DA AND 4)/4;"DRIVE=":DA AND 3
2370 PRINT"N=";HEX$(PEEK(AD+1));" SC=";HEX$(PEEK(AD+2));" GPL=";HEX$(PEEK(AD+3))
;" D=";HEX$(PEEK(AD+4))
2380 AD=AD-2:GOTO 2060
4999 'INVALID
5000 PRINT"INVALID ":PRINT"ST0=";HEX$(PEEK(AD)):AD=AD+1:RETURN
50000 'write machine
50010 READ A$:IF A$="end" THEN RETURN
50020 POKE AD,VAL("&H"+A$):AD=AD+1:GOTO 50010
50030 RETURN
50040 'trans data
50050 READ A$:IF A$="end" THEN RETURN
50060 E=USR1(VAL("&H"+A$)):GOTO 50050
50070 RETURN
50080 'COM-TRANS
50090 DATA F5,3E,0F,D3
50100 DATA FF,F1,CD,7F,27,7D,F5,DB,FE,E6,02,28,FA,3E,0E,D3,FF,F1,F5,D3
50110 DATA FD,3E,09,D3,FF,DB,FF,E6,04,28,FA,3E,08,D3,FF,DB
50120 DATA FE,E6,04,20,FA,F1,C9,end
50130 'RECEIVE
50140 DATA CD,7F,27,CD,12,CC,77,CD,12,CC,CB,47,C2,9C,27,2B,18,F1
50150 DATA 3E,0B,D3,FF,DB,FE,E6,01,28
50160 DATA FA,3E,0A,D3,FF,DB,FC,F5,3E,0D,D3,FF,DB,FE,E6,01
50170 DATA 20,FA,3E,0F,D3,FF,F1,c9,end
50180 'trans data
50190 DATA 50,00,00,0D,DB,F1,CD,0F,07,DB,F3,CD,0F,07,C3,00,50,end
```

FM-7

DISKバックアップ・ツール "TUKBA-7"

オートモードだけど、その強さは…?!

坂本耕二

ＦＭ-７用のバックアップ･ツールを制作しましたので、紹介します。

この DISK バックアップ･ツールは、ディスク・アナライザーの機能は持っていません。オートモードでマスター・ディスクを落とすものです。

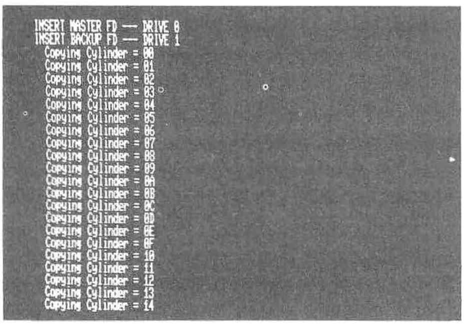
本プログラムの概要は、マスター・ディスケットを１トラックづつ読み、それに基づきバックアップ・ディスケットをフォーマットし、その後の WRITE を行なうものです。

データの WRITE には、256バイトづつ行なっているので現在のところ残念ながら高速とはいえません。(コピー時間は約15分かかりる)

プログラムの入力方法

このバックアップ・ツールは、マシン語だけで作られていますから、リスト１のマシン語をＦＭ-７のモニタあるいはチェックサム付きモニタにより入力します。

プログラムの開始番地は、&H6500番地となっていますから、入力ミスのないことを確認後、BASIC の EXEC &H6500、またはモニタから G6500 でプログラムを走らせることができます。



本プログラムの実行画面。
下の表示が全トラックによって表示される。

使い方

このプログラムは、２ドライブ専用となっていますので、表示のメッセージに従いマスター・ディスケットをドライブ０に、バックアップ・ディスケットをドライブ１に入れます。後は、ひたすらバックアップ・エンドのメッセージが出るまで待ってください。

表示されるトラック・ナンバーは、バックアップの中のトラックを示していますが、16進数になっていますので、&H27トラックが10進数の39トラックとなります。

現在、このバックアップ・ツールは、バージョンアップ中なので、ソフトによってはバックアップできないものがあります。

ただし、以下の条件のプロテクトであればバックアップできると思います。

♥セクタ数が16以外。
♥デリーデット・マークが使ってある。
♥ ID フィールドだけのセクタがある。
♥トラック、サーフェース、セクター・レン

グスの値が異常である。

♥ DATAフィールドに次のIDフィールドがある。

近々、本プログラムに加え、さらに強力化したいバックアップ・ツールを発表する予定です。ご期待ください。

なお、本プログラムの最後にDISK解析用ダンプリスト・プログラムを載せておきます。次回は、このツールをもとにDISKプロテクトの解析を行なうつもりです。次号が発刊されるまでに入れておいてください。

〈リスト1〉プログラム・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
6500  8E 69 8C CC 20 00 ED 03 CC 24 00 ED 05 ED 07 CC  :01
6510  45 00 ED 09 86 0A A7 02 30 8D 04 38 BD 68 F1 BD  :40
6520  DB 54 81 0D 26 F2 30 8D 04 0B BD 68 F1 BD DB 54  :A3
6530  81 0D 26 F2 7F 69 8B 86 80 B7 FD 1D BD 68 FF 17  :2B
6540  00 1B B6 69 9C B7 69 9B 5F 44 59 4F 17 00 17 BD  :C7
6550  68 FF 86 01 17 00 0F BD 68 FF 16 01 1E 7F 69 9C  :F1
6560  86 4F B7 69 9D 39 1A 50 8A 80 B7 FD 1D 4F BA 69  :82
6570  8E B7 FD 18 12 12 12 12 12 12 12 12 B6 FD 1F 85  :41
6580  40 27 F9 39 1A 50 8A 80 B7 FD 1D 86 50 20 DF 1A  :CD
6590  50 8A 80 B7 FD 1D 86 D4 B7 FD 18 B6 FD 1F 85 40  :E8
65A0  27 F9 10 8E 00 00 86 C0 B7 FD 18 31 21 B6 FD 1F  :F4
65B0  2A 07 B6 FD 1B A7 C0 20 F2 12 12 12 12 85 40 27  :AC
65C0  EA B6 FD 18 A7 C0 10 AF C1 33 47 31 22 12 12 12  :9F
65D0  12 10 8C 37 60 23 CF 81 10 26 02 33 46 33 50 C6  :B2
65E0  10 6F C0 5A 26 FB 39 1A 50 8A 80 B7 FD 1D 86 80  :3E
65F0  B7 FD 18 B6 FD 1F 2A 07 B6 FD 1B A7 C0 20 F4 85  :9D
-----------------------------------------------------------
Sum:  4F CD B0 99 09 78 8B 57 D1 31 39 4A 1D 41 A8 B8  :0B

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
6600  40 27 F0 FF 69 99 39 1A 50 8A 80 B7 FD 1D 86 E0  :3C
6610  B7 FD 18 10 8E 00 00 B6 FD 1F 2A 0F B6 FD 1B A7  :EA
6620  C0 31 21 10 8C 19 00 22 06 20 EC 85 40 27 E8 FF  :CE
6630  69 97 39 1A 50 8A 80 B7 FD 1D 86 A0 34 04 AA E0  :66
6640  B7 FD 18 B6 FD 1F 2A 07 A6 C0 B7 FD 1B 20 F4 85  :9D
6650  40 27 F0 B6 FD 18 85 40 39 1A 50 8A 80 B7 FD 1D  :65
6660  86 F4 B7 FD 18 B6 FD 1F 2A 07 A6 C0 B7 FD 1B 20  :9E
6670  F4 85 40 27 F0 B6 FD 18 85 40 39 FE 69 8F 7F FD  :0B
6680  1C 86 00 17 FF 09 10 8E 00 00 4D 4D 4D 4D 4D 4D  :2D
6690  31 3F 26 F6 30 8D 02 76 BD 68 F1 B6 69 8B BD 68  :A6
66A0  F8 FE 69 8F 86 00 17 FE E6 7F 69 9E FE 69 91 86  :73
66B0  00 17 FF 53 17 00 50 17 00 85 24 0A 7C 69 9E B6  :D3
66C0  69 9E 81 05 25 E6 FE 69 93 86 01 17 FF 8B 17 01  :D2
66D0  E3 B6 69 9B B1 69 9D 10 27 02 0E 7C 69 9B B6 69  :3A
66E0  9B 5F 44 59 C1 01 26 05 F7 FD 1C 20 B4 4F 17 FE  :CC
66F0  93 86 01 17 FE 8E 7F FD 1C 7C 69 8B 20 96 7F 69  :63
-----------------------------------------------------------
Sum:  50 9C 1E C8 36 53 1B BB 4E 74 61 19 4E 5D 5A E7  :59
```

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
6700	9C	86	4F	B7	69	9D	39	FE	69	95	BE	69	8F	EC	07	83	:8F
6710	00	00	27	28	A6	06	81	08	26	06	86	FF	A7	09	20	15	:1A
6720	A6	84	B7	FD	19	A6	02	B7	FD	1A	EF	0A	4F	17	FE	B7	:81
6730	B6	FD	18	A7	09	30	88	10	EF	0A	20	D1	EF	0A	39	BE	:1D
6740	69	8F	FE	69	91	A6	C0	81	F5	26	02	86	75	81	F6	26	:8C
6750	02	86	76	81	F7	26	02	86	77	A7	5F	11	B3	69	97	26	:8B
6760	E4	FE	69	91	A6	C0	11	B3	69	97	10	24	00	A2	4D	27	:50
6770	06	81	FF	27	13	20	ED	A6	C0	4D	26	E8	A6	C0	4D	26	:67
6780	E3	A6	C0	4D	26	DE	20	12	A6	C0	81	FF	26	D6	A6	C0	:14
6790	81	FF	26	D0	A6	C0	81	FF	26	CA	10	8E	00	10	EC	C0	:A6
67A0	10	83	A1	FE	27	12	10	83	A1	F8	27	4F	10	83	A1	FB	:3C
67B0	27	49	31	3F	26	E8	20	AC	31	51	C6	0C	6F	A0	5A	26	:9D
67C0	FB	CC	F5	F5	ED	5D	A7	5F	EC	84	ED	41	EC	02	ED	43	:BD
67D0	EC	04	81	F5	26	02	86	F7	81	F6	26	02	86	F7	C1	F5	:DD
67E0	26	03	CC	F7	00	C1	F6	26	03	CC	F7	00	C1	F7	26	03	:70
67F0	CC	F7	00	ED	45	30	88	10	16	FF	69	31	51	C6	0C	6F	:FE
Sum:	C1	D6	1B	4D	E3	0D	80	F9	34	88	DB	42	6B	21	F2	F1	:B0

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
6800	A0	5A	26	FB	CC	F5	F5	ED	5D	A7	5F	31	41	16	FF	54	:FC
6810	BE	69	8F	EC	07	83	00	00	26	05	1C	FE	16	00	94	A6	:C1
6820	09	85	10	26	02	8D	09	10	25	00	88	30	88	10	20	E3	:E4
6830	FE	69	91	11	B3	69	97	25	03	1A	01	39	EC	C0	10	83	:77
6840	F5	F5	26	EF	EC	41	10	83	F5	FE	26	E7	EC	43	10	A3	:A1
6850	84	26	E0	EC	45	10	A3	02	26	D9	11	B3	69	97	24	D3	:2A
6860	EC	C0	10	83	F5	F5	26	F2	A6	41	81	F5	26	EC	A6	42	:98
6870	81	FE	26	03	1A	01	39	33	43	10	AE	0A	5F	10	AC	88	:DD
6880	1A	24	1E	A6	A0	81	F5	26	04	86	75	C6	01	81	F6	26	:A1
6890	04	86	76	C6	01	81	F7	26	04	86	77	C6	01	A7	C0	20	:B4
68A0	DC	A6	09	85	08	26	08	86	F7	A7	C0	86	4E	A7	C4	EA	:53
68B0	09	E7	09	39	BE	69	8F	EC	07	83	00	00	27	2A	A6	09	:5E
68C0	81	FF	27	1F	85	01	27	1B	85	08	26	17	5F	85	20	27	:83
68D0	01	5C	A6	84	B7	FD	19	A6	02	B7	FD	1A	EE	0A	86	01	:49
68E0	17	FD	50	30	88	10	20	CF	39	30	8D	00	39	BD	68	F1	:60
68F0	39	34	76	BD	9B	DB	35	F6	34	76	BD	AC	3D	35	F6	34	:F0
Sum:	20	4D	CB	39	8E	2F	BF	10	A9	89	83	20	DF	36	6D	26	:7A

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
6900	14	C6	03	8E	FF	FF	30	1F	26	FC	5A	26	F6	35	94	0D	:26
6910	0A	20	20	43	6F	70	79	69	6E	67	20	43	79	6C	69	6E	:42
6920	64	65	72	20	3D	20	00	0D	0A	43	6F	70	79	20	45	4E	:1D
6930	44	20	21	21	20	00	0D	0A	49	4E	53	45	52	54	20	42	:14
6940	41	43	4B	55	50	20	46	44	20	2D	2D	2D	20	44	52	49	:C4
6950	56	45	20	31	00	0C	0D	0A	49	4E	53	45	52	54	20	4D	:51
6960	41	53	54	45	52	20	46	44	20	2D	2D	2D	20	44	52	49	:CF
6970	56	45	20	30	00	0D	0A	43	6F	70	79	69	6E	67	20	54	:4F
6980	72	61	63	6B	3D	20	00	00	00	00	00	00	00	00	00	00	:FE
6990	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
Sum:	66	EC	F8	78	AA	08	59	74	DF	0C	62	26	3A	58	46	3E	:CA

●セーブの方法／SAVEM"COPY", &H6500, &H6985, &H6500↵

●実行／LOADM"COPY"↵　EXEC &H6500↵（またはオートスタート）

メッセージに従いマスターをドライブ0に入れコピー用ディスクをドライブ1に入れる

DISK解析用ダンプリスト・プログラム

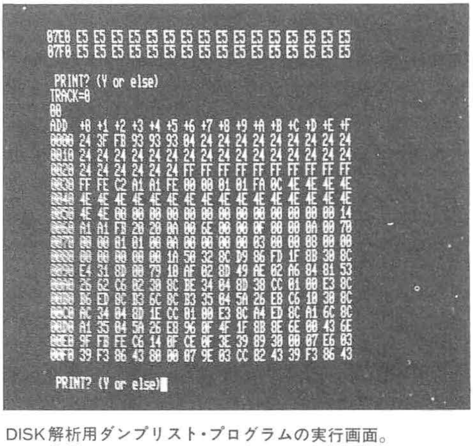
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
6000	1A	10	BD	DA	F6	86	80	B7	FD	1D	7D	FD	1D	2A	F6	7F	:C4
6010	FD	1C	8E	80	00	86	D4	B7	FD	18	30	8D	01	43	BD	9B	:A6
6020	DB	BD	DB	54	35	10	81	59	26	03	7C	05	AC	34	10	30	:B0
6030	8D	01	44	BD	9B	DB	7F	61	83	BD	DB	54	81	0D	27	25	:2E
6040	91	30	25	F5	81	3A	24	F1	34	02	80	30	78	61	83	78	:65
6050	61	83	78	61	83	78	61	83	BA	61	83	B7	61	83	35	02	:0C
6060	BD	D0	8E	20	D4	BD	9B	50	B6	61	83	84	F0	44	44	44	:91
6070	44	C6	0A	3D	34	04	B6	61	83	84	0F	AB	E0	81	50	24	:36
6080	99	34	02	81	28	24	03	4F	20	02	86	01	B7	FD	1C	35	:9C
6090	02	81	28	25	02	80	28	B7	FD	1B	BD	AC	3D	86	10	B7	:3C
60A0	FD	18	F6	FD	18	54	25	FA	8E	80	00	7F	FD	0F	86	E0	:92
60B0	F6	FD	18	54	25	FA	B7	FD	18	F6	FD	1F	2A	07	B6	FD	:40
60C0	1B	A7	80	20	F4	58	2A	F1	BF	61	81	8E	80	00	7D	FD	:F2
60D0	0F	34	10	30	8D	00	53	BD	9B	DB	35	10	BD	9B	50	34	:B7
60E0	10	30	89	80	00	1F	10	BD	AC	37	35	10	86	20	BD	D0	:90
60F0	8E	5F	7F	FD	0F	A6	80	7D	FD	0F	BC	61	81	10	27	00	:FC
Sum:	C8	67	6F	E2	C9	79	3E	32	90	52	80	53	53	BB	4F	1B	:5F

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
6100	27	BD	AC	3D	86	20	BD	D0	8E	5C	C1	10	26	E4	1F	10	:F4
6110	5D	27	02	20	C7	BD	9B	50	B6	03	13	7F	03	13	81	40	:37
6120	26	AF	7F	05	AC	16	FE	EA	16	FE	E7	0D	0A	41	44	44	:DE
6130	20	20	2B	30	20	2B	31	20	2B	32	20	2B	33	20	2B	34	:91
6140	20	2B	35	20	2B	36	20	2B	37	20	2B	38	20	2B	39	20	:AA
6150	2B	41	20	2B	42	20	2B	43	20	2B	44	20	2B	45	20	2B	:F1
6160	46	00	0D	0A	20	50	52	49	4E	54	3F	20	28	59	20	6F	:79
6170	72	20	65	6C	73	65	29	00	0D	0A	54	52	41	43	4B	3D	:2D
6180	00	98	C9	00	00	00	00	00	00	00	00	00	00	00	00	00	:61
Sum:	CD	D7	E8	53	19	29	4D	E1	37	38	DD	91	1A	64	D3	BF	:3C

●セーブの方法
SAVEM"TDUMP", &H6000, &H6183, &H6000↵

●実行
LOADM"TDUMP"↵
EXEC &H6000↵
または、オートスタート

☆最初にプリンタの使用を聞いてくるので使用時は"Y"を、それ以外であれば印字しない。
次にダンプするトラック・ナンバーを10進で入力。(0→79)



DISK解析用ダンプリスト・プログラムの実行画面。

DISK バックアップ・ツール カタログ



PC-8801/mkII/SR用 DISK COPY TOOL

EXPERT 88

(エキスパート88)

定価 12,800円

SOFT PAL

〝コピー・ツールにEXPERTあり！〟とハッカー仲間には評判の高い製品だ。オートモードあり、アナライズ機能ありと、強力なプロテクトをいろいろな切り口で解析し、落せるのが特徴。主な機能としては、

1. Disk Analyzer：DISKの解析用ツール。これでプロテクトの状態を見る。また自分のソフトにプロテクトをかけることも可能。
2. コンストラクション・モード：バックアップ（オート）モードで落せないソフトのために、専用のプログラムを用いバックアップを取るいわゆるファイラー・モード。
3. バックアップ・モード：オート・モードのツール。Hand Pick B1にも勝るとも劣らないレベルだ。

以上の他に立ち上がり時間と短縮できるインタリーブ・フォーマットやドライブ・スピード・チェック機能などを持つ。

なお、上記の2については、「EXPERT NOTE」でサポート。



PC-8801/mkII/SR用 ディスク解析用ソフト

RATS & STAR

(ラッツ・アンド・スター)

定価 12,800円

Atarix

DISKアナライザー＋ファイラー・モードのバックアップ・ツールだ。オート・モードの他にも、プロテクトを自ら解析し専用の簡易言語を用いて落とすか、または、同社が発行している〝パラメータ・マガジン〟（DISK版もある）に載っている各ソフト専用のファイラー（パラメータ）で落とす方式。

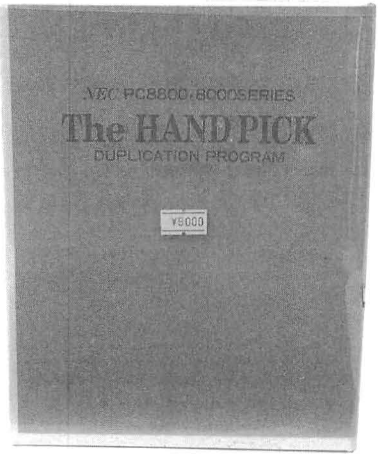
●主な特徴

解析ツールと呼ぶに相応しくディスクの内部をのぞき、分析するためには充分すぎるほどのツールが用意されている。

また、解析したデータを元に簡単な操作で内容の書き換えができるという、上級GAMEハッカーにも手応え充分な製品。

初心者には、先ほどのパラメータ・マガジンを用いバックアップを取れば良いだろう。なお近々、バージョンUPを行なうとの話もあるようだ。

●SOFT PAL/〒556 大阪市浪速区日本橋4－7－22池友ビル303 ☎06－644－3782
●Atarix/〒113 東京都文京区本郷2－40－9小林ビル5F ☎03－816－4841



PC-8801/mkII/SR
PC-8001/mkII/SR用
DUPLICATION PROGRAM

The HAND PICK B1

(ハンドピックB1)
定価 9,800円
Sofmap

最強のオートモード・ツールとして今年初めに発売されたのが、〝HAND PICK B1〞。各ドライブのスピード・チェック、マルチウインドウ方式のトラック状態表示、ステータス表示、メニュー表示など、最強の名をはずかしめない内容を持っている。

肝心なバックアップ・ツールとしての力であるが、発売されて以来、HAND PICK をしのぐ製品は出現していないほどだ。

しかし、最近の新作ソフトのプロテクトは、ありとあらゆるテクニックを駆使してあり、すでにオートモード1本槍では落せないものがほとんどである。

とはいっても、オリジナルのフォーマットそのままでバックアップが取れるオートモード方式は魅力的であり、ぜひ本製品のバージョンUP版を望みたい。

耳より情報……

RATS&STAR FM のVer2.0が発売になります。現時点では入手できませんが、概要がわかりましたのでお知らせします。

まず、他のコピー・ツールと比べても割と強力なXX COPYが改良・強化されてMaster Copyとなりました。Master Copyはオートモードなので、だれでも簡単に使えます。オートモードが強化・高速化されることによって必要なパラメータも少なくなるでしょう。

最近、多くなってきた4ドライブにVer2.0から対応するようになりました。5インチ・ドライブと3.5インチ・ドライブが接続してあればメディア変換ができます。

ディスク・コンペア機能が追加になり、2枚のディスクを比較して違っている場所を調べやすくなっています。これは、COPY BOY-7などの影響があるのでしょう。

RATS&STAR 特有の簡易言語も追加・修正によって多機能化がなされています。なおVer・1.0をお持ちの方は4,000円の交換料で Ver 2.0 にバージョンアップしてもらえるのはうれしいところです。

●〝ファイラーでソフトが落とせない!!〞だって? こんな声が、ハッカー仲間に広がっている。いろいろと聞いてみると、どうら、ソフト・ハウスが同一のソフトでも出荷時期によって、プロテクトの掛け方が変えてあるらしいのです。

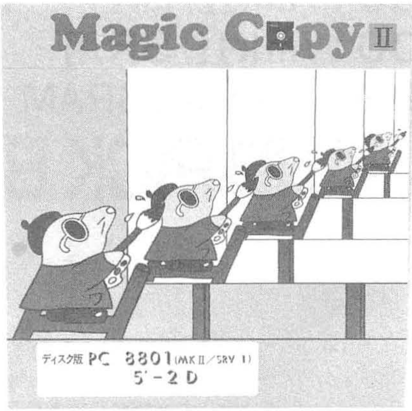
これでは、1つのソフトに幾つものパラメータが必要となってくるわけです。

バックアップ・ツールの発売元では、〝何とかして新しいパラメータの情報をユーザーに知らせたいのだが〞と頭を悩ませているようだ。

また、このようなプロテクトに対応するには、ツール・メーカーの動きを素速くキャッチするしかないだろうが、EXPARTやRATS&STAR等のパラメータ・マガジンを定期的に発刊しているところは、こういった新手のプロテクトには有利に対応できるであろう。

●The BASIC で有名な技術評論社から9月15日に『ザ・ディスク解析ツール』という本が定価2,000円で発売になった。締切り間近のため具体的には紹介できないものの、ナカナカの内容なので、PC98、PC88、FM−7各シリーズのユーザーなら一読の書と言えよう。定価2,000円なら決して高くないので読んでみてほしい。

●Sofmap/〒101 東京都千代田区外神田3−5−4末広町ハイム401 ☎03−253−4226



PC-8801/mkII/SRV1用 ファイラー・モード・ツール Magic Copy II

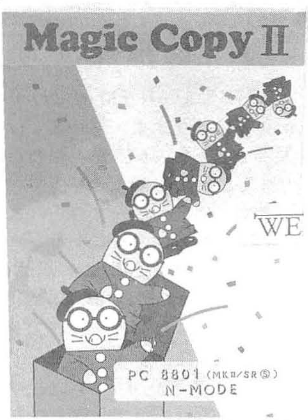
（マジック・コピーII）
定価 9,800円(OPTION PACK 各2,300円)
WESTSIDE ソフトハウス

ノーマル、オートマティック、マニュアルタイプと方式の異なるバックアップ・プログラムを内蔵し目的によってセレクトできる。

ファイラーは、70種以上のソフトに対応したものが始めから組込まれており、それを利用することで簡単にバックアップすることができる。これ以外のソフトについては、オプションパックとして別売されているファイラーを利用すれば良い。

バージョン3.2では、さらにエキスパンド・モードが付加された。このモードは、今までのオートマティックの内容を表示する機能が付いたもので、オートで取れなかった場合の解析資料として利用することができる。

番外編 PC-8801/mkII/SR用 ファイラー・モードのテープ・ツール TAPE版 Magic Copy II



ディスク版ソフトが一般的となった現在でも、テープ版ソフトは、まだまだ多く出回わっている。

最近のテープ・ソフトのプロテクトは、ディスク版の強力プロテクトを反映してか強力化してきている。この強力化してきているテープ・ソフトを何とかバックアップしようと作られたのが〝テープ版MAGIC COPY II〟である。

MAGIC COPY IIは、ファイラーを用いたディスク版バックアップ・ツールとして有名であるが、〝テープ版〟も同様にファイラーを用いてバックアップすることができる。

■ファイラー・モードがある

MAGIC COPY IIの特徴と言えば何と言ってもファイラー・モードがあることだ。オートモードは、そのプロテクトの状態によって4つのレベルを選べるが最近の強力なプロテクトに対抗するためファイラー・モードが付加されているのである。

オートモードは、そのバックアップ方法が分かってしまえばそれ以外の方法でプロテクトを掛けることで対抗されてしまうが、ファイラーはソフト毎にバックアップ方法を指示してあるため、そのソフトのファイラー

●WESTSIDEソフトハウス/〒661 大阪府尼崎市庫之荘1－11－5湯川ビル2F ☎06－436－2799

があれば必ずバックアップできる。ただし、ソフトメーカーもそれほどアホではなく、ロットごとにプロテクトを変えるという対抗策を出してきている。

テープ版マジックコピーIIのファイラーは現在カセットの形で3本出ている。(内容については表1参照) 今後、他のソフトについてもサポートされるであろうから期待したい。(ちょっと欲を言えば、ツールを購入した時点では、ファイラーが1つもなく、別にそのテープを購入しなければならない。オマケ的に数点入っていてもバチは当たらないと思うのだが。)

〈表1〉テープ数マジック・コピーIIのファイラー

現在3本のファイラーが出ている(各¥1000)

ファイラー		
ファイラー1	・JELDA II	・Cannon Ballrun X
	・大脱走	・Applon
	・HUDSONシリーズ	・Galaxian
	・ドアドア	・One Way Trap
ファイラー2	・F2グランプリ	・Dragon Slayer
	・飛車	・キン肉マンProgram A
	・Cosmo Mutar	・キン肉マンProgram B
	・TNT BOMb BOMb	
ファイラー3	・Chack'n Pop	・Newtron
	・HAUNTED CAVE	・王将
	・ワイルド・ウェスタン	・Dream Land
	・SPACE CRUISER	

PC-9801用

BABY MAKER

オートモード+パラメータ
定価14,800円(8インチ、5インチ2DD、3.5インチ)
マイコンシステム

バックアップ・ツールの元祖といえば、このBABY MAKERがあげられる。

まず、オートモードでバックアップ、これで落ちないソフトはパラメータをセットしバックアップするという2段構えの攻めかたができる。

このパラメータ・セットの方法はオートモードでは不可能とされていたμPD765A(PCのFDC用IC)以外のプロテクトにも対応し、市販ソフトの99%が落とせるという。

さらに、ドライブが1~4まで自由に設定できるので、2HDと8インチ、5インチ2DDと3.5インチの変換もお手のもの。

なお、ディスク・アナライザー機能は6つの表示画面で解析でき、各画面モードごとにHELPキーで補足的な機能でアナライズ可能だ。

パラメータ機能で気になるのがサポート体制だが、当社では最新ソフトのパラメータを網羅したディスクを1,000円で提供している。

スーパーバックアップSBU

オートモード
定価49,801円
(メンテなし19,801円)
MAT(エムエイティ)

98のハッカー連中で、強いという評判の高いオートモード・バックアップ・ツールが〝スーパーバックアップSBU〟だ。

以前、欠点とされていたコピー・スピードがアップされ使いやすくなっている。

使い方は、画面に表示されるメッセージに従う方式で、コピー先のディスクはどんなフォーマットでも、元のディスクのフォーマットに変換されるのが特徴。

さらに、アナライザーの機能を備え、各トラックのセンター情報をサーチできる。

なお、定価49,801円のものは、メンテナンス付きで、新バージョンを無料で9ヶ月間サポートしてくれる。

メンテナンスなしは、1回9,000円でバージョンアップしてくれる。

マイコンシステム/〒171 東京都豊島区高田3-16-11 川野ビル4F ☎03-981-0563
MAT /〒799-15 愛媛県今治市国分乙389-3 ☎0898-48-3848



FM-7/NEW7/77 Super Disk Analyzer

RATS&STAR FM

(ラッツ・アンド・スターFM)

定価 12,800円　Atarix

ＰＣ88シリーズで好評のRATS & STARのＦＭ版。オートは３種類あり、ＸＸCopy、AutoCopy,Express Copyの３本でプロテクトの強さに併せて選べる。簡単なプロテクトならば、Express Copyで十分で、スピードも早い。

付属のパラメータは３本と少ないが、パラメータマガジンで新作ソフトのパラメータがサポートされている。パラマガを見て、自分で簡易言語により入力することもできるが、ディスク版も出ているので入力が面倒な人もＯＫ。

取説やパラメータマガジンはプロテクトに関して良い参考書になる。それにしても、コピーするための簡易言語を作ってしまうとは驚き。

Super Disk Analyzerと銘打つだけあって、アセンブラ、逆アセンブラが使えるのはスゴイ!! 初心者にも十分使えるが、マニア受けするツールと言えよう。

タイトル画面がイマイチなのでカットした方が良いのでは……。



FM-7/77 DISK COPY TOOL

EXPERT FM

(エキスパートFM) Version3.1

定価 12,800円
SOFT PAL

Ver 2.5がバージョンアップされて、現在発売になっているのがVer 3.1だ。オートはあまり強くないが、コンストラクションモードがあり、46本の専用コピープログラムが内蔵されているので、多くのソフトに対応できる。

新しいコンストラクションは"EXPERT NOTE"にてサポートされているので、新しいプロテクトにも追従できる。88版も出たのでサポートに期待したい。

どのツールでもそうだが、同じFM-7でも５インチでは動作するが、3.5インチでは動作しないということが多い。3.5インチ→５インチはＯＫでも、５インチ→3.5インチはダメということも多い。その点、EXPERT FMは3.5インチ専用のコンストラクションがあったりでうれしい。

FM-77でも完全に動作するコピーツールの市販をツール屋さんにお願いしたいものだ。

同じVer 3.1でも旧タイプの取説より最近の取説はかなり詳しくなっており、市販ソフトのプロテクトの実際などが紹介されている。

●メディアポリス/〒150 東京都渋谷区神南１－11－５ダイネス壱番館10Ｆ ☎03－463－0411

DISK バックアップ・ツール カタログ



FM-7/NEW7/77用
DISK保守•統合ユーティリティ
COPY BOY-7
(コピーボーイ7)

メディアポリス　定価 9,800円

COPYBOY-7 の Auto　Copy はナカナカ強力で新しい製品だけに他のコピーツールではオートで落ちないものも落としてくれる。Auto Copy で落ちない場合はアナライザーを使ったり、パラメーターもサポートされているのが安心だ。

ただ、付属のパラメータは3本しかなく、今後のサポートが気になるところ。パラメータが少ないのはオートが強い証拠とも言えるが……。

パラメータはプロフェッショナル麻雀、Dr.COPY　2、EXPERTFM の3本。ちなみに、プロテクトが変わったのかパラメータでも EXPERT FM ver 3.1 は取れなかった。

初心者にはオート&パラメータ、上級者にはアナライザー&エディタと使えるうれしいソフトだ。

F-BASIC を拡張した COPY-BASIC も搭載されているので技術があれば自分でパラメータを作ることも可能。

また、COPYBOY-7 以外のコピーツールをチェックしたり、ディスケットのチェック、ドライブのチェックなどの機能もあり使いやすい。



FM-7/NEW7/77用
TAPE ⇨ DISK
留年生 (RYUNENSEI)

定価 7,800円
BIG HEAD

ＰＣ60用のツールに浪人生というのがあるが、こちらは〝留年生〟というタイトル。なぜ、こんな名前なのか考えてしまった。どうやら「留年生＝落ちる」で落ちるにひっかけているらしい。

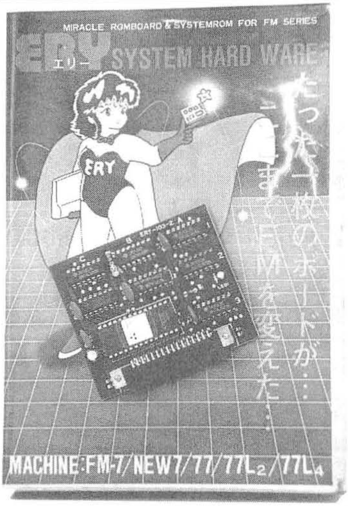
ここでは、Tape to Disk 版を使ってみた。テープ版のソフトをディスクに落とすツールというわけだ。取説には28本のメディアコンバート方法が載っている。

最近、ディスク・ドライブを手に入れたという人には、テープソフトをディスクにメディア変換できるので便利なツールと言える。慣れてくれば、取説に載っていないソフトでも落とせるようになってくる。

なお、Disk to Disk 版もあるので、ディスクユーザーの方はそちらも利用してみてほしい。

Disk に4本しか入らないツールもあるが、本ソフトの場合は F-BASIC と同様に152本のプログラム（最大）が1枚のディスクに入れられるのが特徴。価格が7,800円と安いのもユーザーにとってはありがたい。最後に、なぜか本ソフトにはプロテクトがかけられていなかった。

●BIG HEAD/〒380 長野県長野市柳町76 ☎0262−33−1911



FM-7/NEW7/77/77L2/77L4用 ROMボード&システムROM

ERY(エリー)

定価 19,800円 ミラクルシステムサービス

本ボードは、他とは違ってハードウェアをFM-7/77各シリーズのオプションスロットに差し込むもので、ボード上のROMからプログラムを読み込むため、ソフトは不要。

どちらかと言えば、テープユーザー向けに便利なボード。ボード本体はコンパクトにできているので、FDインターフェースや漢字ROMカードなどよりも小さく、電源スイッチをONにするとLEDが点燈する。

RAMファイルとして使用できるほかに、F-BASICが拡張され、高速LOAD/SAVE (CMT) が可能になる。F-BASICの約8倍のスピードなのでテープ待ちのイライラも解消される。

その他、NEW復活やアンリスト解除などのコマンド追加、拡張モニタなどの機能が追加されている。ただ、FM-7の場合はFDインターフェースや漢字ROMカードなどが入れられていると使用できないのが残念といえる。できれば、ハード版だけでなく、ソフト版の方も出してもらいたい。

なお、本ボードはコピー・ツールではないので注意されたい。



FM-7/NEW7/77用 オートモード・コピー・ツール

ロリコピーFM

定価 9,800円 ソフトハウス・パス

ロリコピーはVer 1.0→Ver 2.0→Ver 2.1とバージョンアップされて現在に至る。Ver 1.0からオートモード1本のツールだ。他のツールがファイラー、パラメータ、コンストラクションなど専用のプログラムをサポートしているのに対しオートモードのみなので古いソフトにかなり対応するものの新しいソフトだと苦しくなる。エラーメッセージが日本語で表現されるのはわかりやすいが……。

他のコピーツールでもロリコピーをオートで落とせるものが出てきたり、ファイラーでサポートされたりするものが多くなった。弱いコピーツールだと、〝ぶ～とえら～〟の表示が、出てくるのでコピーツールの強さの目安になるだろう。コピーできたように見えても、途中で止まってしまう場合もあるので注意したい。

プロテクトが弱い場合は1度に4トラック分をコピーしてくれるのでコピー時間は早いが、プロテクトが強くなると何度もやり直すので時間がかかる。

他のツールやアナライザーと組み合わせて使う必要がありそうだ。

●ミラクルシステムサービス/〒700 岡山市奉還町1－9－4 和泉ビル2F ☎0862－53－7920
●ソフトハウス パス/〒[redacted]

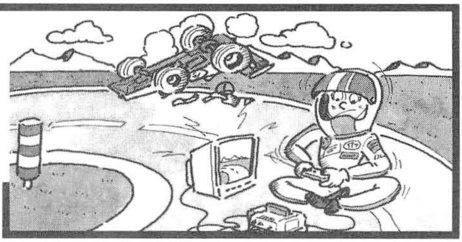
初公開

第2章 ファミコンの徹底解剖&改造テクニック

丹治佐一



Part〈1〉
ファミコンの徹底解剖
ファミコンは日本のアップル



任天堂のファミコン、すごいですねー。へたなパソコンよりずっと面白いゲームがあるし、そのスピードと画面のきれいさではパソコンよりすぐれています。

パソコンでゲームをしようとしている人は、パソコンを買わずにファミコンにしよう!!そうすればパソコンを買うための資金のほとんどをROMカセットというゲーム・ソフトの購入に当てられるのでスピードも遅く、くだらんパソコン・ゲームを1〜2個〝オマケ〟でもらうより、よっぽど楽しめます。〝ゲームはファミコンだ!〟

僕の親指には、十文字の〝ゼビウス・ダコ〟や〝スパルタンXダコ〟があります。それほど引き付ける楽しさがあるのです。(パソコンはくさるほどあるのに）これと同様に楽しむことのできるパソコンと言えば、米国はアップル社のアップルII/Plus/II e/II cでしょうか。

ゲーム・ソフトの多さと質の高さから言ったらまだ米国にはかないません。(あのロードランナーやチョップリフターなどは、アップル用に作られ走っていたものです）

〈表1〉仕様(一部)

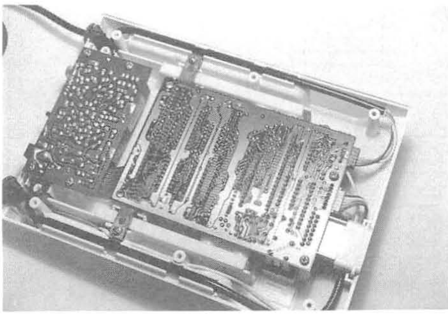
●LSI	CPU：Nchカスタム 8ビットCPU、クリスタル21.47727MHz 6502 ソフトウエア・コンパチ PPU：ビデオ・コントローラ Nchカスタム ハードウェア・スクロールが可能なため、 スームーズなスクロールを実現 RAM：2Kバイトスタティック RAM(2016)×2⇨4Kバイト実装 ROM：(カセット内) プログラム 最大32Kバイト（256Kビット マスクROM) キャラクタ（キャラジェネ）最大8Kバイト（64Kビット マスクROM)
●グラフィック	色 ：52色 分解能：横256×縦240ドット
●電源	電圧：DC10V 850mA 供給、内部では5Vレギュレータ I Cにより +5Vで動作
●コントローラ	転送方式：コントローラ内にパラレル→シリアル変換 I Cを持ち、 キー内容はシリアル（直列）で本体に送られる

ファミコン大解剖

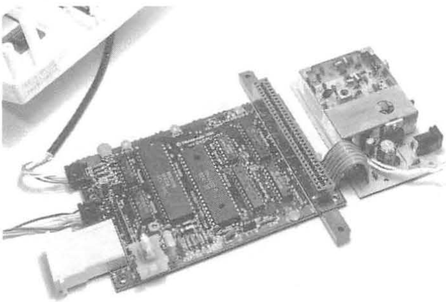
あのおもしろいゲームは、ファミコンのどの部分から生れてくるのでしょうか。ファミコンを解剖して調べてみることにしましょう。

●本体●

ファミコン本体の裏ブタのネジをはずすと基板が現われます。(写真1)基板は2枚に分かれていて、マイコン部分と電源、RF信号を作り出す部分とがフラット・ケーブルで結ばれています。(写真2)



〈写真1〉ファミコン内部



〈写真2〉部品面より

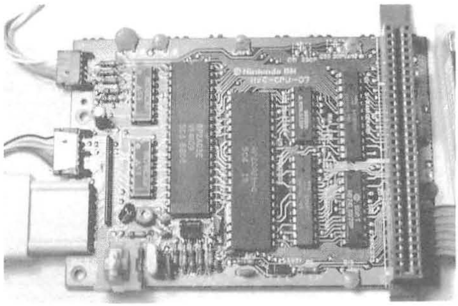
ファミコンで使われているCPUはカスタム（特注品）であるため、ピンコネクションなどは不明ですが、マシン語（ソフト・ウェア）レベルはアップル社のアップルでおなじみの6502 CPUと同じです。

これを見ても、ファミコンが日本のアップルだ！と言えますね。

6502 CPUは8080 CPUの流れにあるZ80 CPUとは別のCPUで、6800 CPUの流れの中にあります。6800と言えばFM-7に使われている6809が有名です。

6502とソフトウェア・コンパチですからアップルを持っていて、〝マシン語なんかバリバリ〟だったりすると、〝ファミコンのゲーム・ソフトを作ってください〟と依頼が来るとか来ないとか……。

話を元に戻して、CPUの型番にRP××と付いているところを見るとリコーが作ってい



〈写真3〉基板アップ

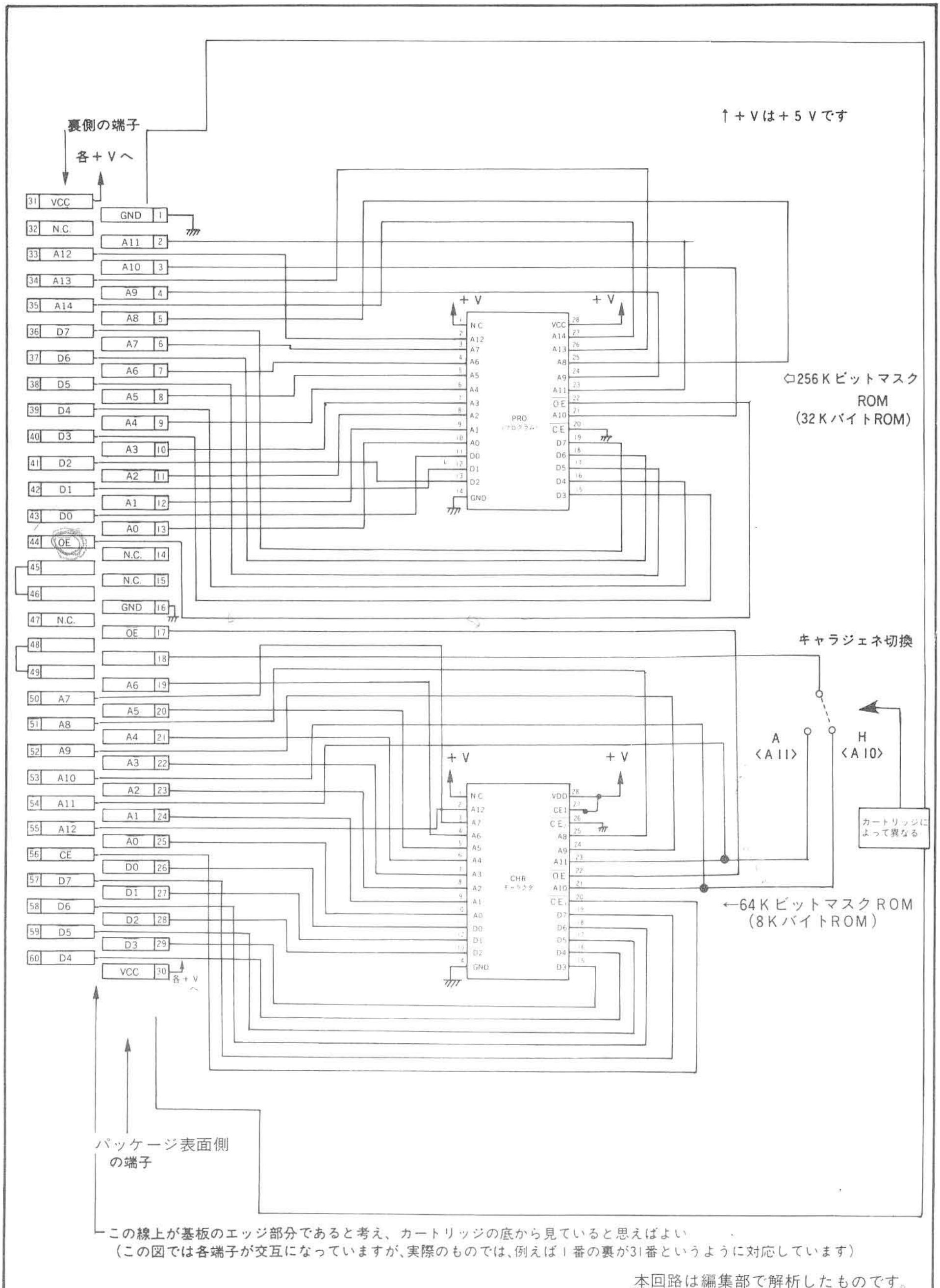
るのではないかと思われます。

PPU：このビデオ・コントローラでは、あの、MSXで使われているTMS 9918といったVDPのようにスプライト動作ができるようです。それに、画面を上下左右に動かすスクロールは、ソフト・ウェアによって行なっているのではなく、ハード（PPU）によって行なっているため、とてもなめらかなスムーズ・スクロールが可能になっています。

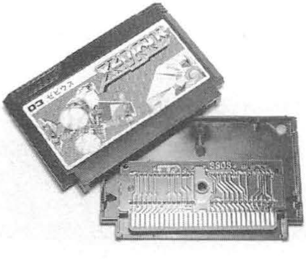
ですから、ゼビウスやロードランナー、ポパイでブルートが下に落ちた時などのリアルな上下動など、ソフトだとぎこちない動きがスムーズに動いているわけです。

RAMは、本体中の4Kバイトのみです。PPUの中にメモリを持っているかもしれませんが、プログラムやキャラジェネのメモリは不要だと言ってもCPUのワーク・エリアを入れて4Kバイトしか使えないのですから、プログラムのたいへんさがわかります。

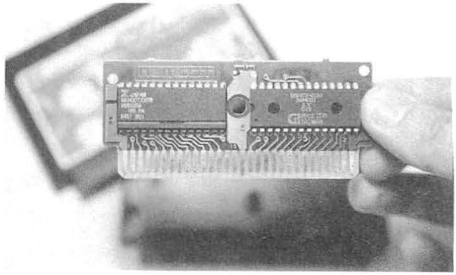
●ROMカセット●



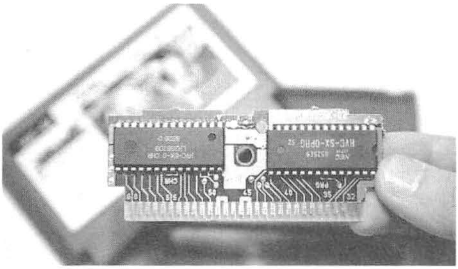
〈図1〉ROMカセット配線図　この図は〝スパルタン〟のものですが、キャラジェネ切換とメモリ容量の違い以外、各端子の接続は同じです



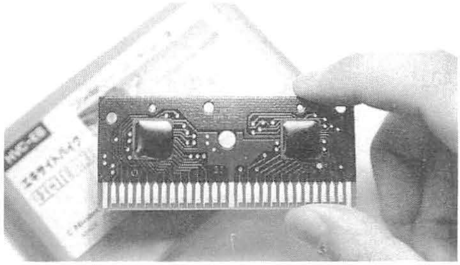
〈写真 4〉 カートリッジ内部



〈写真 5〉 基板アップ(ゼビウス)



〈写真 6〉 基板アップ



〈写真 7〉 エキサイトバイクROM

プログラムが入っているカセットは、プログラム用に最大32Kバイトのマスク R O M(256 Kビット)とキャラクタ用に最大8Kバイトのマスク R O M(64Kビット)が入っています。

任天堂のエキサイト・バイクなどは、基板上にR O Mがくっついた形でできており、クロック・モジュールの基板を見ているようです。他のメーカー製では、マスクR O Mが使われています。このマスクR O Mというのは、書き込むプログラムやデータといった内容をI Cを製造する段階で作ってしまうもので、後からプログラムをI Cに書き込むより安く作れます。ただ個数が多くまとまらないとだめなんです。

メーカーによって、マスクR O Mのかわりに後から書き込んだり、消法できる EPROM を使っている場合があるようです。

●電源・RF部●

ファミコン内の電源部は、外部からのDC 10Vを内部で使う＋5Vに安定化させています。もし自分で外部電源を作るときは、7V以上の電圧を加えるようにしてください。＋5V1AのレギュレータICは、定電圧化させる電圧（5V）より＋2Vぐらい高い電圧がないとだめだからです。

CPU側からの出力信号（音声、画像）はフラット・ケーブルでこの基板に入り、RF信号に変換されます。つまり、入る前段は、RF信号ではなく、ビデオ信号で入力されているのです。これを利用すればビデオ出力が取り出せそうですね。

基板から、製作しているメーカーはアルプス製であることがわかりました。



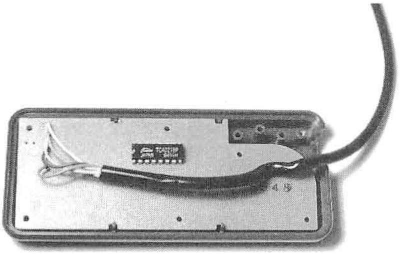
〈写真 8〉 レギュレータICのアップ

●ジョイスティック●

ファミコンで使われているジョイスティックは、MSXなどのジョイスティックとは異なり、そう単純ではありません。

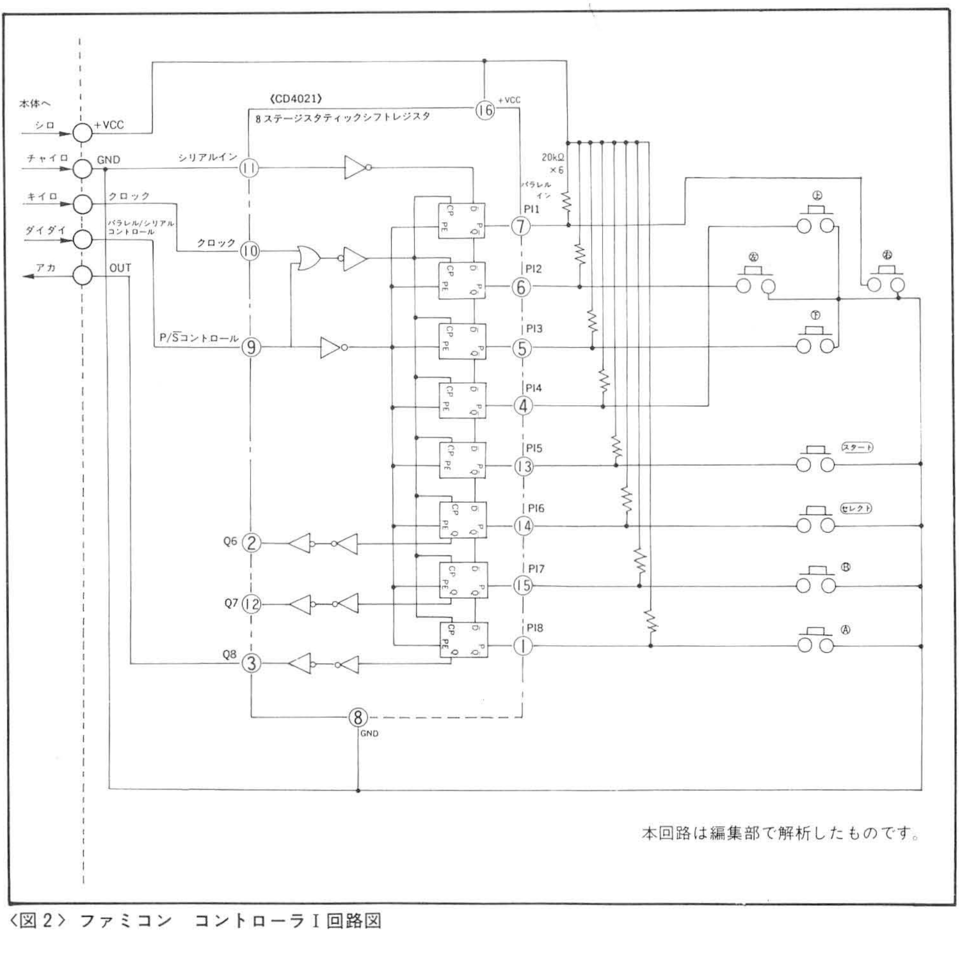
MSXは、キーの内容をそのまま伝える並列方式ですがファミコンでは、すべてのキー内容を一本の信号線で送るシリアル（直列）方式を使っています。

キー内容（並列）から信号線（直列）にする役目は、ジョイスティック内のIC（CD4021：8ステージ・スタティック・シフトレジスタ）によって行なわれています。



〈写真9〉ジョイスティックIの内部

ファミコンからジョイスティックまでの線は5本（コントローラIIではマイク出力があるため6本）あり、＋5V、GND、出力の他



〈図2〉ファミコン　コントローラI回路図

に、パラレル/シリアルの切り換えに一本、そして各キー内容を同期させて1つずつ送り出すためのクロック信号に一本使われています。

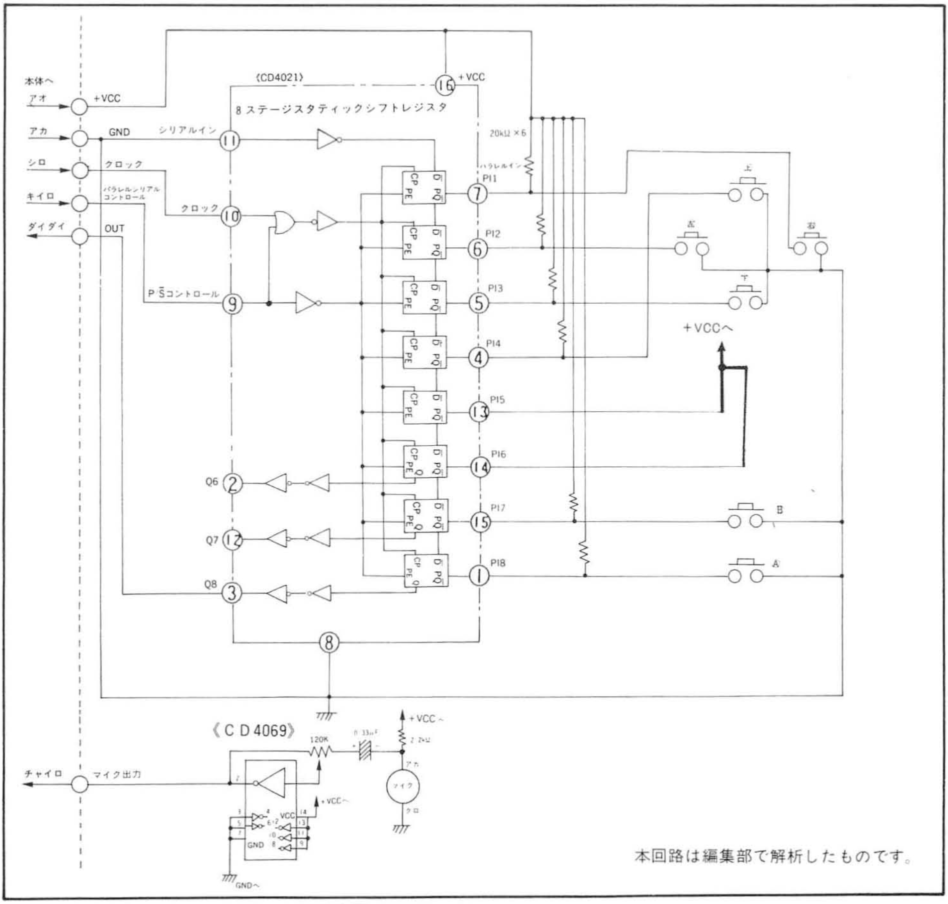
コントローラIIでは、Iに付いているスタート、セレクト・スイッチ部分のIC入力ピンが直接GND（アース）に落ちています。さらに、IIのみの機能であるマイク入力は、4069というC-MOSインバータをリニア的に動かしマイク・アンプとして利用しています。

コントローラIIの基板を外すと、スライドボリュームの一部が基板上に作られているため、ボリュームとして用をなさなくなり、その状態で電源をONにすると発振しっぱなしで動作しなくなってしまいます。注意／

●グラフィック●

ファミコンのグラフィックは今のパソコンのグラフィック（640×200または640×400ドット）から比べると、そうドット数のある細かい点が出せるグラフィックではありません。

横256×縦240ドットですから、1つ1つのドット（点）はきれいに出ているようです。



〈図3〉 ファミコン コントローラII回路図

640×200ドットといった縦長のドットから考えればドット数は少ないものの、ひずみの少ないグラフィック画面が得られているわけです。

色は52色で、1つのキャラクタ・パターンは8×8ドットで作られています。そして、1つのキャラクタに対して2枚のパターンが重なった形で使われています。

これは、キャラクタ自身に点のある、無し以外の色指定も行なわせているためです。つまり、2枚の同じ8×8ドットが重なっているとすれば、各点のあり、無しから最大4色（点ナシ入れて）をキャラクタ・パターンから指定できることになります。

あとは、アトリビュートなどで各パターン毎の色を指定していると思われます。(1つのキャラクタ・パターンで最大4つまでの色しか指定できないということになります)

1つのキャラクタが8×8ドットですから、これを文字数にすると、ファミコンでは、(256×240ドット)最大32文字×30行表示可能ということになりますね。

Part〈2〉ファミコン改造 Ver.1

スパルタンX、ゼビウスには装備したい『ハイパージョイスティックの製作』

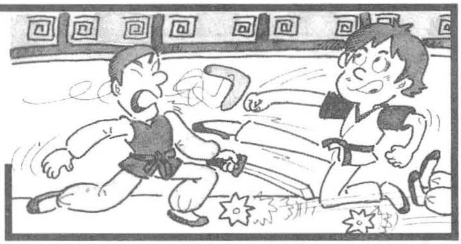
親指に十文字のゼビウス・ダコを作ってしまうというのも、ついゲームに力が入ってしまうからなんですが、A、Bといった発射ボタンを素早く押さなければ数多く玉が発身されなかったりスパルタンXでは、すぐ抱き付かれてしまうため、方向キーを駆使せざるをえないためです。

そこで、ゲーム必勝のためのハイパー回路を取り付けることにしましょう。

ハイパー回路の効用

何発も発射させたい時は、普通コントローラのIやIIのA、Bボタンを素早く押したり放したりを繰返さなければなりませんが、このハイパー回路を搭載することで、その必要は、まったくなくなります。

連打したくなったらAまたは、Bボタンをただ押し続ければ良いのです。もうこれで勝ったも同然だ！



ハイパー回路の効用は、と言いますと、〝スパルタンX〟ではもう楽！相手がやって来る前にパンチやケリを入れておくと勝手に相手が死んでゆきます。

2～3発打ち込まないと死なない相手は、相手のそばまで行って、ただボタンを押すだけでOK！知らないうちに2～3発入って相手は死んでゆきます。〝ゼビウス〟では、ハイパーがなくても押し続けることで連打してくれますが、ハイパーを搭載すればもっと高速に連打させることができます。

これにより弾幕なんか簡単に張れるのです。きりもみ回転してつっ込んでくるジアラも簡単にやっつけられる。ただ難点と言えば始めから飛来するものをすべてやっつけてしまうと、〝ハイレベルのプレーヤーだ〟と思い込まれ、自動的に〝スーパーゼビウス〟のモードになってしまうのです。

スタート直後から普通は出てこないものまでがバンバン飛来してくるわけです。まあ、腕に自身のある私には関係のないことですが。

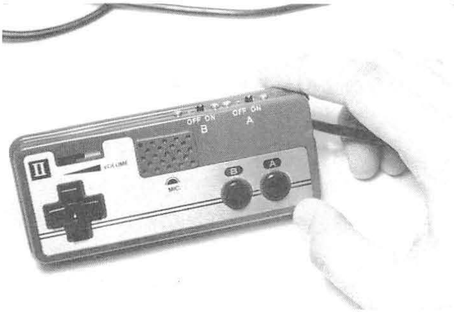
〝テニス〟では、カッコいいサーブが必ず決められます。玉を持った時点でＡボタンを押していると、玉を投げ上げた瞬間パカーンとサーブを打っています。プロのテニス・プレーヤーがやるサーブと同じワザが使えるんです。また、連打モードのままでテニスをすると、数打ちゃ当たるで〝プレーヤーの体力〟を心配しつつ、ただ単にボタンを押し続ければ良いというこの安易さが最高です。

その他、ギャラクシアンでは、敵に当たり出すとピシュン、ピシュンと気持ちよいリズムで相手を消してゆけるし、800点は必ず取れます。などなど、まだありますが、それは使ってのお楽しみ。

ハイパー回路のしくみ

前書きが長くなってしまいました。さっそくハイパー回路について説明しましょう。

ハイパー回路は、ジョイスティックのＩま

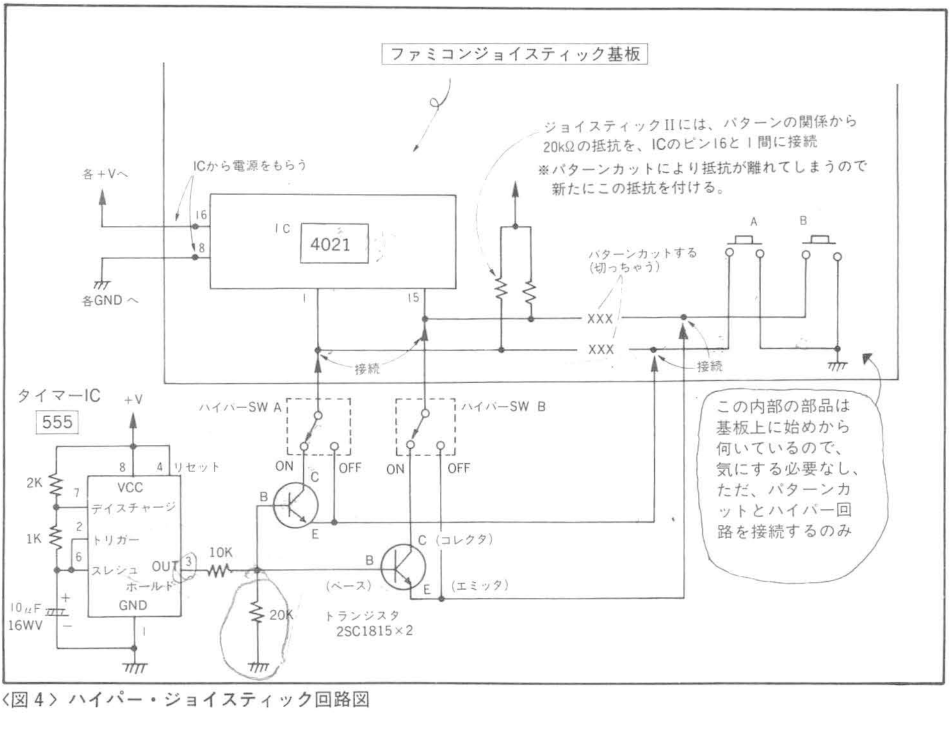


〈写真10〉 スイッチ部

たはIIに内蔵します。そして、ボタンＡまたはＢにその回路を付加にするかのスイッチを付けることになります。

普通スイッチＡ、Ｂを押しっぱなしにするとＯＮのままであるため、１回押したとしかプログラム上では見てくれません。

そこで､ジョイスティックのスイッチＡ(Ｂ)とＩＣとの間に別のスイッチを付け（パターンカットしてスイッチを直列に入れる)､その



〈図４〉 ハイパー・ジョイスティック回路図

スイッチを高速でON/OFFしてやれば、たとえスイッチA（B）が押しっぱなしであっても回路の途中が断続するため素早く連打していることになります。

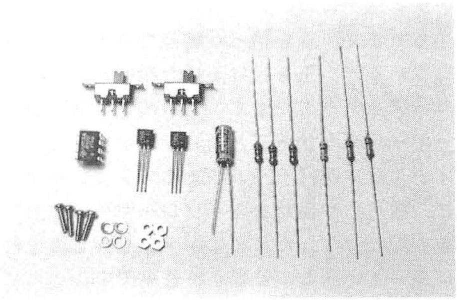
この直列に入れるスイッチの役目をトランジスタに行なわせ、そのON/OFFのコントロールは、タイマーＩC555のパルス波を利用します。

連打させるスピードは、555の出すパルス波の周波数によりますが、スイッチの内容を読むのはソフトであるため、読むタイミングと連打するタイミングとが合わなければ、いくら高い周波数のパルスを発振させてON/OFFさせても押してないと判断されてしまいます。今回の回路定数は、実際にゲームをしながら発振周波数を変えて、最も高速で連打してくれるときの値を用いています。

また、連打したくないゲームなどがあります。例えば〝ドルアーガの塔〟では、剣の出し入れを繰返してしまいます。そこで、トランジスタをパスするスイッチをスイッチA、B毎に付け好みに応じてハイパーを利用できるようになっています。

ハイパー回路を作ろう

まずは、表2の通り部品を集めましょう。タイマーＩC 555はポピュラーなICなので半



〈写真11〉 ハイパー部品

導体屋さんへ行き〝555〟くださいと言えば話は通じます。トランジスタも2SC1815以外に2SC945など汎用のトランジスタでOKです。ハイパーON/OFFで使う切り換えスイッチはファミコンのジョイスティック内に組み込むために薄型のスライド・スイッチを使います。ON/OFFのみで1回路用のものを捜してください。

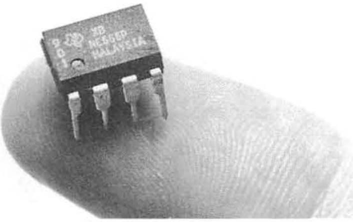
●ＩC部の製作●

買ってきたＩCの状態では、ジョイスティック内に入れることはできませんから、ラジオ・ペンチなどを使ってＩCの足を平らに延ばして、そこに部品を取り付けていくことになります。（写真12）

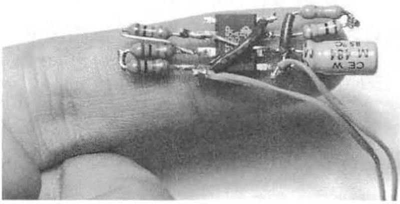
できるだけ薄く小さくなるように気を付けながら部品を取り付けていきます。この時、各

・ＩC	タイマーＩC NE555	1
・トランジスタ	2SC 1815など同等のもの	2
・抵抗	1 kΩ1/4w（茶黒赤金）	1
	2 kΩ1/4w（赤黒赤金）	1
	10 kΩ1/4w（茶黒橙金）	1
	20 kΩ1/4w（赤黒橙金）	2
	└各ワット数は1/4wのもので1/8wのように小さなものを選ぶ	
・コンデンサ	電解 10μF 16WV（10WV以上であれば良い）	1
・スイッチ	ON/OFF 2Pスライドスイッチ（一回路）うす型で小さいもの	2
・リード線		少々
・スイッチを取り付けるためのビス、ナット（φ2 mm×5 mm）		各4

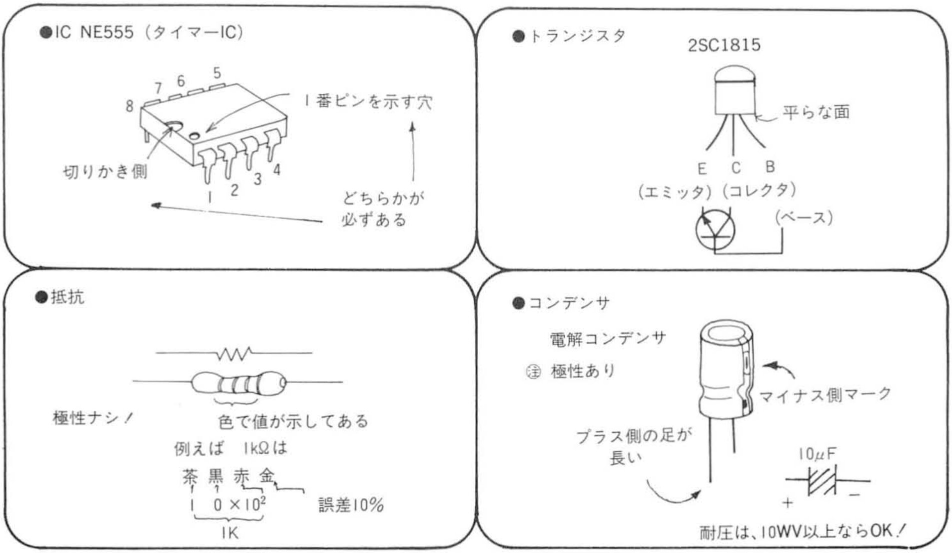
〈表2〉 ハイパー・ジョイスティック部品表（ジョイスティック1個の部品）



〈写真12〉 555のアップ



〈写真13〉 ハイパー回路のアップ



〈図5〉 部品図

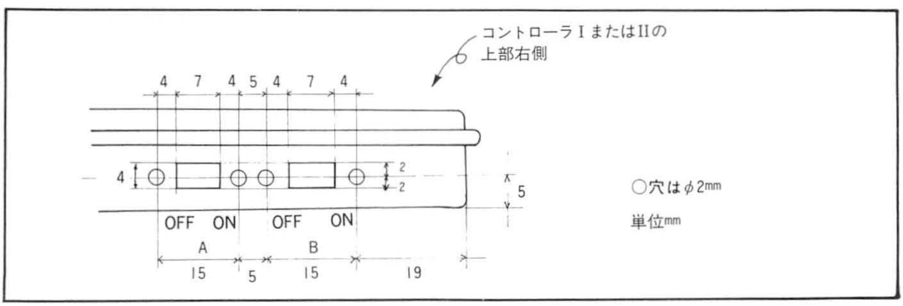
配線は、部品の足を使い、交差するような部分には、絶縁チューブを付けます。

（写真13を参照）

●スイッチの取り付け

ハイパーON/OFFのスイッチを付ける穴をあけましょう。穴のあけ方は、穴の位置をマークしたあとハンド・ドリルで直径3 mmぐらいの穴を2ヵ所ほどあけ、カッター・ナイフでマークした位置まで広げていきます。ジョイスティックで使われているプラスチックは以外に柔らかく簡単に穴を広げることが

〈図6〉ハイパー切換スイッチ加工穴位置



できました。

スイッチの取り付けネジ穴は、スイッチを取り付けた状態で穴を開ければ位置の間違いもなく正確に穴があきます。

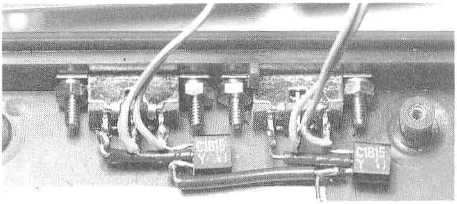
●組込み●

穴が開けられたら、そこにスイッチを取り付けトランジスタをハンダ付けします。(写真14) そこに、ＩＣの底に両面テープを付けたハイパー回路の抵抗とトランジスタとをハンダ付けしドッキングさせて完成です。(写真15)

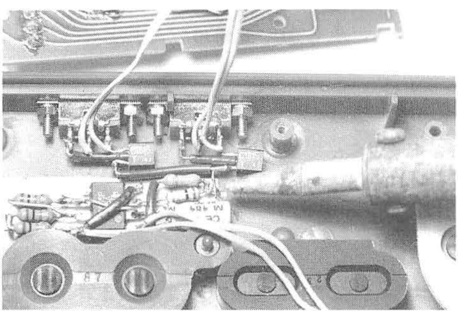
次に、ジョイスティック基板のパターンこの一部をカットし、そこにハイパー回路からの出力を接続します。パターンの関係からジョイスティックIIには抵抗20kΩをＩＣ(4021)のピン16とピン１間に取り付けます。(写真16)

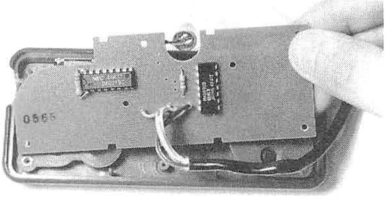
接続部分は、写真17、18と19、20を参考にしてください。

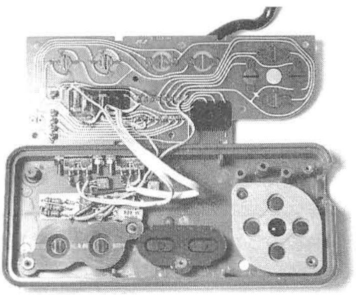
基板をネジ止めする前に、絶縁テープなどハイパー回路の上や基板等で貼り絶縁しておきます。(ＩＣのピンが飛び出ている場合は、ニッパ等などで切り取って絶縁します)

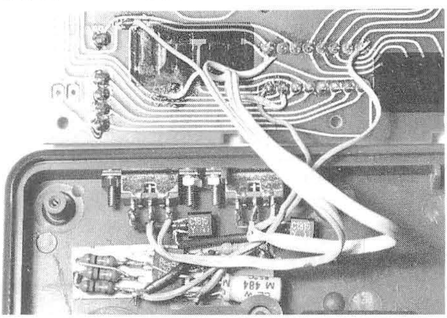


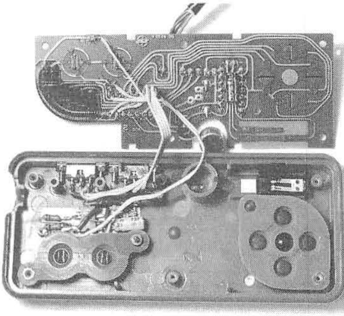
〈写真14〉 トランジスタのアップ

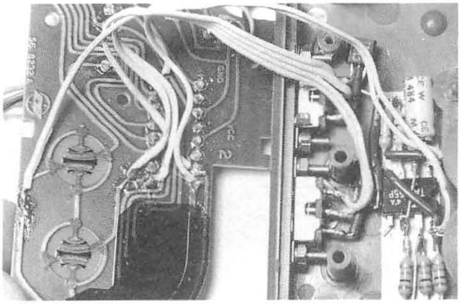

〈写真15〉 Trとハイパー回路をドッキング


〈写真16〉 ジョイスティックIIの内部


〈写真17〉 ハイパー回路ジョイスティックＩの全体


〈写真18〉 ハイパーの超アップ


〈写真19〉 ハイパーの全体、ジョイスティックII



〈写真20〉 ジョイスティックIIの超アップ

使い方

使い方は、とても簡単。ハイパーで戦いたいときは、使いたい側の切り換えスイッチをONにするだけです。あとは、普通の通りゲームをすればOK／

うまい使い方は、例えば〝ギャラクシアン〟では、A、Bスイッチともに玉を発射できることを利用して、片側のAならA側のハイパーONにしておけば普通時はAを使ってバカ打ちをし、危険な時など、必ず消さなければならない際に、ハイパーOFFのBを使う(逆の考えも成り立つ) などの方法があります。

使い込むとわかるのですが、〝ゼビウス〟などは、〝弾幕を張れる〟とつい強気になってしまい、自ら敵に体当りしてしまうというアホなことをやってしまいます。くれぐれも〝ハイパーの使い過ぎには注意しましょう。

Part〈3〉
ファミコン改造 Ver.3

〝キャンプ〟DE〝ファミコン〟
取っておきの ビデオ出力化
ポータブル化 ファミコン電源



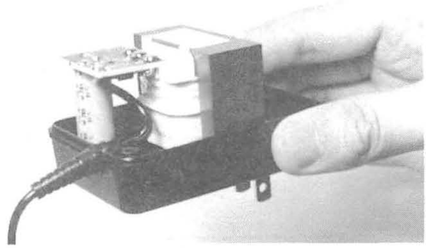
家の中でファミコンをやっているなんて暗い／これからは〝アウト・ドア・ファミコン〟の時代ですよ。というわけで都会の雑踏から離れ、自然の空気の中でキャンプをしながらファミコンを楽しむ。ちょっと贅沢な遊びのようだけど以外に簡単に実現できてしまいました。これからは、空気も澄んで紅葉も見ごろ、みなさんも〝キャンプDEファミコン〟をやってみませんか？

ファミコン用外部電源

アウトドア・ファミコンでネックとなるのはやはりファミコンを動かすための電源とTVでしょうか。発電機を持っているという方はまずいないでしょうから、バッテリーで動作させる



〈写真21〉 バッテリー・ボックス全景



〈写真22〉 ACアダプタ内部

ようにします。

ファミコンのＡＣアダプタは10Ｖとなっていますが、ファミコンの内部は、レギュレータＩＣにより＋５Ｖで動作しています。つまり、外部電源は、あえて10Ｖは必要ないですし、逆に言えばレギュレータＩＣが入っているため車のバッテリー（12Ｖ）から電源をもらっても良いわけです。ただ入力電圧が12Ｖとなると出力電圧＋５Ｖとの差がすべて熱として出てしまうことになります。

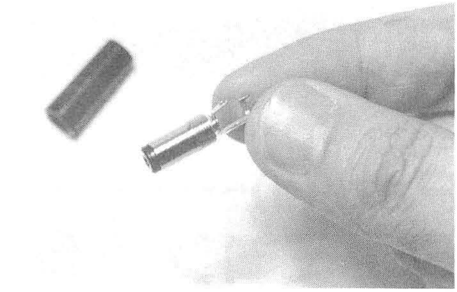
そのため、レギュレータＩＣがチンチンに熱くなるわけで、使っていてあまり気持ちの良いものではありません。そこで＋12Ｖラインにワット数の大きな抵抗を直列に入れて使いましょう。

車のバッテリーが使えない場合は、電池をつないでバッテリー・ボックスを作りましょう。

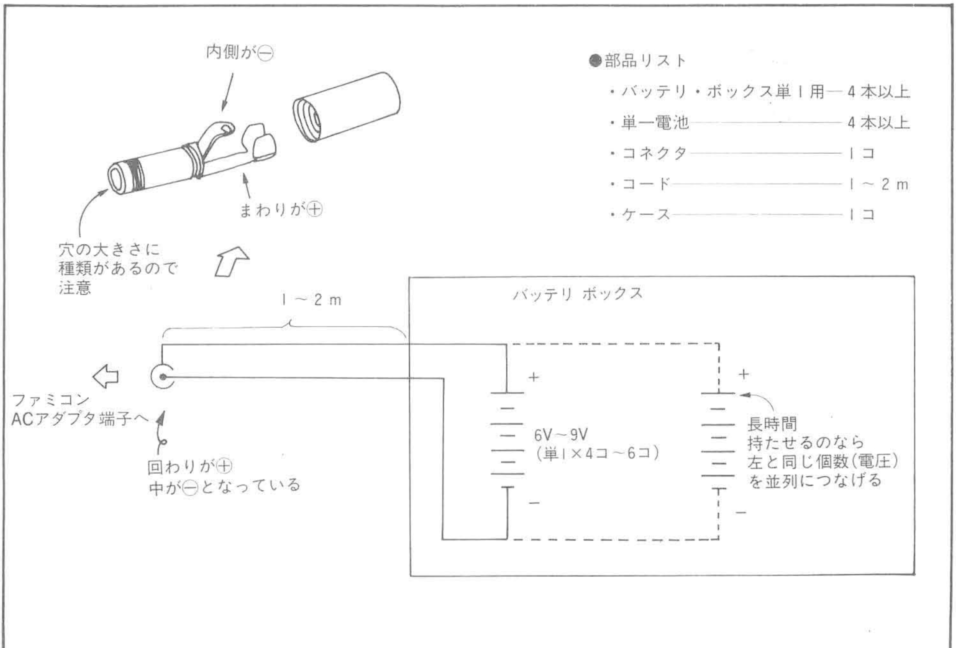
電流を食うので持たせるために単一電池を



〈写真23〉 バッテリー・ボックス回路図



〈写真23Ａ〉 プラグのアップ。外側にプラスがくるよう配線

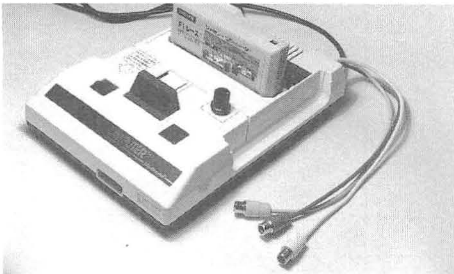


〈図７〉 ファミコン・バッテリー・ボックス回路図

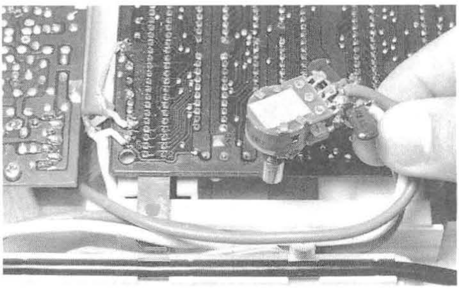
使い、６Ｖ～９Ｖの電圧（単１×４～６本）の一組を並列に並べて電源にするのがベターでしょう。今回は、単１×４本で６Ｖのみのバッテリー・ボックスを作ります。(写真23、図7)

電源製作時に注意したいのは、電源コネクタの電圧の極性です。ファミコンはまわりが⊕内側が⊖になっています。

ビデオ出力化



〈写真24〉後からビデオ、サウンド出力が出ている



〈写真24A〉 VRが表に出ている

ファミコンのＴＶ出力はＲＦ出力であるためにどうしてもノイズっぽい画面になっています。パソコンではビデオ出力やＲＧＢ出力といった信号を直接モニタＴＶに入れているためノイズ感もなくピシッと出ています。

ＲＦ信号はビデオ信号（映像信号）をＴＶの2ｃｈや1ｃｈといったアキチャンネルの電波に乗せ、それをＴＶ側で再び電波から映像信号を得るというようにムダな手間があるため、直接映像信号を入れるものより鮮明感がなくなってしまうのです。

そこで、ファミコンからもビデオ信号を取り出して、きれいな画面でアウトドア・ファミコンを楽しみましょう。

●ビデオ出力の取出し方●

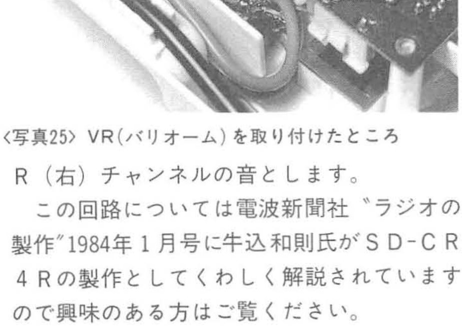
ファミコンには２枚の基板が入っていると書きましたが、１枚の大きな基板は、マイコン基板で、もう１枚の小さな基板とフラットケーブルでつながれています。実は、小さな基板は、電源部とＲＦ信号にするためのＲＦコンバータ部なのです。つまり、この基板へ入力されているのはＲＦ信号ではなく、ＲＦ信号にする前のビデオ信号なんです。

ですから、このフラット・ケーブル部からビデオ信号をもらえば良いわけです。

ファミコンのステレオ・サウンド化

さて、このフラット・ケーブルからは、オーディオ信号も取り出せます。(取り出したオーディオ信号は高域が伸びとてもきれいな音)

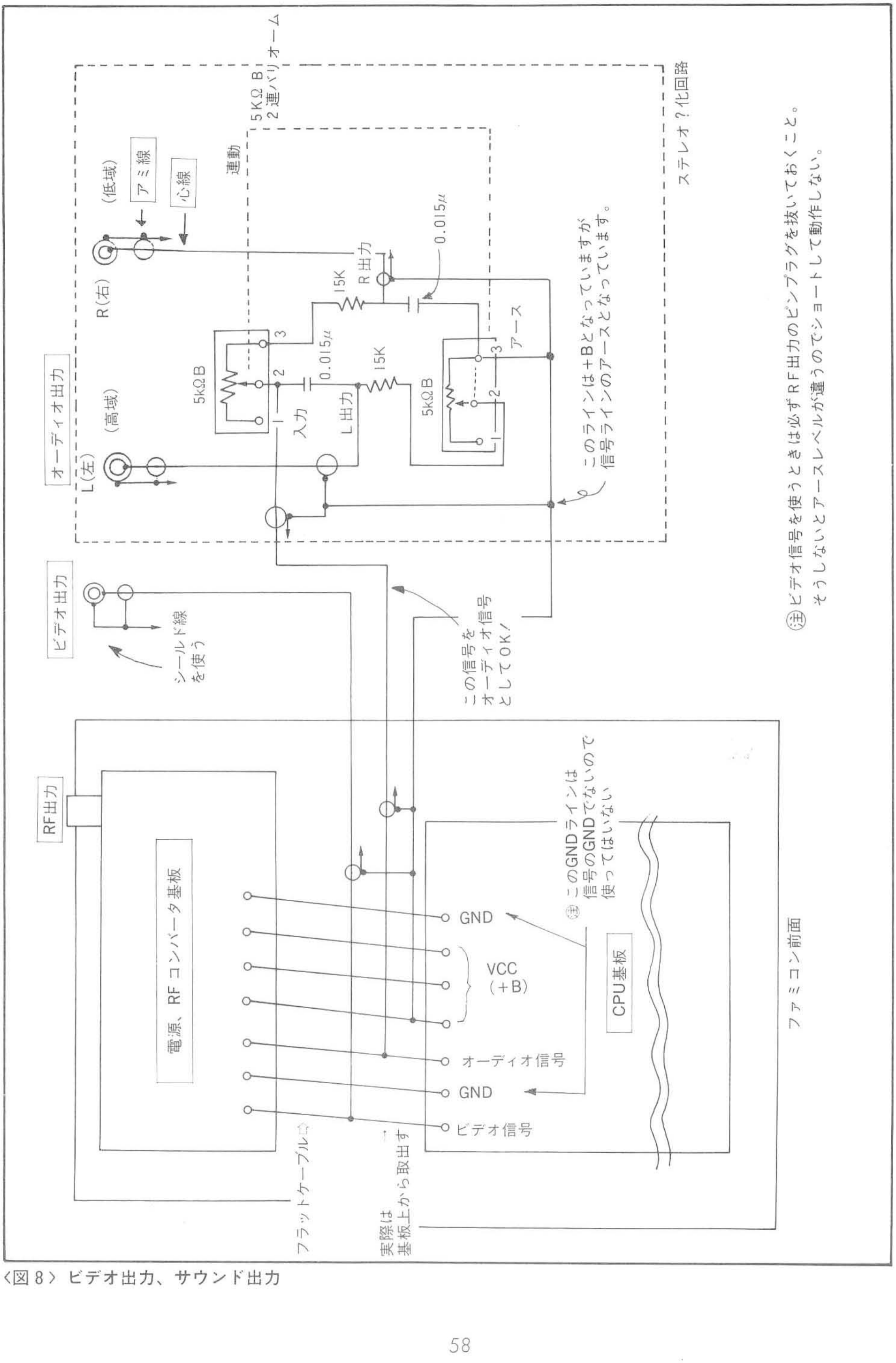
取り出し方は、図８、９図を参考にしてください。ところでオーディオ信号は、モノラルでニューメディア対応のＴＶに接続しても、ちっとも面白くありません。そこで音を高域成分と低域成分とに分け、それをＬ（左）と



〈写真25〉 VR(バリオーム)を取り付けたところ

Ｒ（右）チャンネルの音とします。

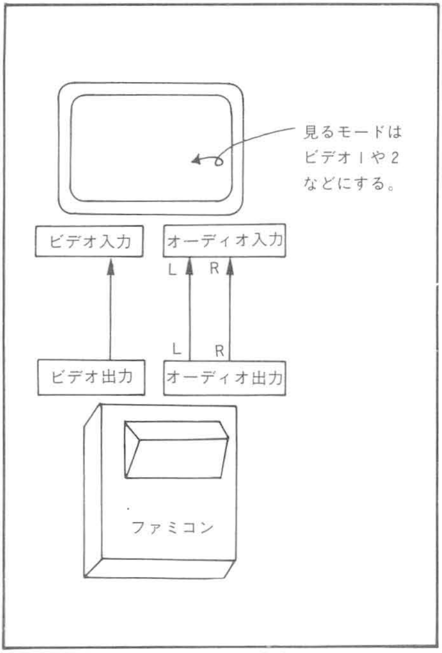
この回路については電波新聞社〝ラジオの製作″1984年１月号に牛込和則氏がＳＤ-ＣＲ４Ｒの製作としてくわしく解説されていますので興味のある方はご覧ください。



〈図 8〉ビデオ出力、サウンド出力

- シールド線　（3色）　赤、白、黄色 ……………………………… 各1m
- RCAピンジャック（メス）　赤、白、黄色 ……………………………… 3
- 抵抗　15kΩ　1/4w ……………………………… 2
- バリオーム　5kΩB　2連　（小型のものでないと入らない）アルプス製 ……………………………… 1
- コンデンサ　マイラ　0.015μF ……………………………… 2
- ツマミ ……………………………… 1

〈表3〉ビデオ出力、ステレオサウンド化部品表



〈図9〉メディア対応TVへの接続

この回路を取り付けゲームをしてみるとおもしろい、F1レースではエンジン音がステレオ感覚で聞え、追い越すときのピュゥーン…という音が左から右へと移動するのです。

ツマミの使い方は、左右のバランスを変えるもので、音を聞きながら好みの位置にしてください。(ファミコンなどコンピュータ・サウンドは高域成分が多いため、どうしても音が左側に片寄ってしまいます。そこで、ステレオ対応などのTVでは左右の音量を調整して使ってください)

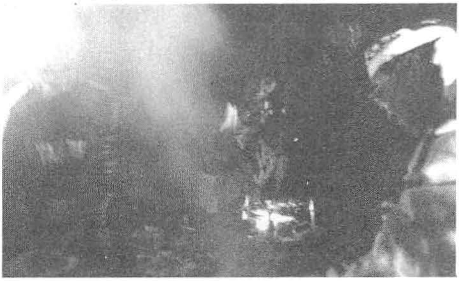
さあ、キャンプに行くぞ

バッテリーもできたことということで、ファミコン用のカラーTV(バッテリー動作可能)を持参し、キャンプ場へと向いました。

キャンプ場は、某所プライベート・キャンプ場です。都心？から高速で30分という近さにありながらこんな所があったのかと思う場所で川が流れ、最高の場所です。(この場所は、僕らが勝手に決めたキャンプ場なのです)



〈写真26〉仕事を忘れて〝今夜は最高!!〟



〈写真27〉オイ！焼鳥が火事だぞ

夜食は、焼鳥、ギョウザ、焼そばと酒の肴ばっかりでしたがビール片手に今夜は最高！

仕事に来たんだと言い聞かせ、ファミコンをと思ったところ〝We are the world〟など



〈写真28〉メットとゴーグルはエキサイトバイクの必需品

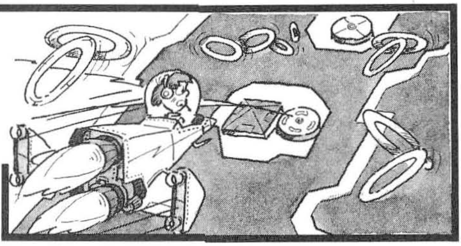


〈写真29〉キャンプDEファミコンスタッフ一同

を見ていてＴＶを明け渡してもらえず、240発のロケット花火と打ち上げ花火により一日は終わりました。次の日、朝食のあと実行しようと思ったところ、持ってきたはずの〝バッテリーがない！ＡＣアダプターもない〟と仕方なく、別のＡＣアダプターを改造しやっとのことでファミコンが楽しめました。

次回のキャンプは、必ずバッテリーを持っていくぞ！

Part〈4〉ファミコン改造 Ver.4



皆んな知ってるかな『ファミコン隠しコマンド』

ちょっと一息入れましょう。というわけでファミコンの〝隠しコマンド〟についてお話ししましょう。

ゲーム・ソフトの中には、取説に書いていない〝隠しコマンド〟を持っているものがあります。このコマンドは、実行すると、別の遊び方ができてしまうものです。この種のものは、ナムコがお得意で、〝ナムコでなければやらない〟という感じがすごくあります。

ゼビウスの隠しコマンド

ゼビウスの隠しコマンドを実行することで当っても死なない〝ソルバルウ〟にすることも、相手の強さ、スタート時の〝ソルバルウ〟の数などを設定することもできるのです。これは、ゲームセンター並の機能なのです。ボード上でスイッチによって設定することと同様なことがファミコンでもできてしまいます。

コマンドの実行方法

ゼビウスを入れ電源ＯＮにしたあと、〝XEVIOUS〟と表示しているデモ画面中にコントローラIIを操作して設定します。

操作は、ＡまたはＢボタンを押しながら右、上、左、下キーの順番で９、２、２、９回押していけばＯＫ！画面右上に８ケタの数が出ればコマンドは受け付けられたことになります。（図10）次に、左右キーで赤くなる数字の位置を変えて、Ａ、Ｂボタンで数字を０→１、１→０にして設定します（４表）。

設定が終わったら〝リセット〟を押して、いつもの通りゲームをすれば、設定したようにゲームが実行できます。

死なない〝ソルバルウ〟にするには、８ケタの数の一番左側の数を１にすればＯＫ！〝リセット〟を押してスタートで恐いものなし！

隠しコマンドで表示される８ケタの数字（０と１の集まり）

0	0	0	0	0	0	0	0

⇦ 電源ON時はすべてゼロとなっている

▲（1ケタ目）　▲３ケタ　▲２ケタ　▲２ケタ

■ ソルバルウの数を設定

４通り
- 00 —— ３機
- 01 —— １機
- 10 —— ２機
- 11 —— ５機

■ 敵の強さを設定

４通り
- 00 —— 普通(何もしないときのモード)
- 01 —— ちょっと簡単
- 10 —— ちょっと強い
- 11 —— 強い(スーパーゼビウスモード？)

■ ソルバルウを一機増すための得点を設定

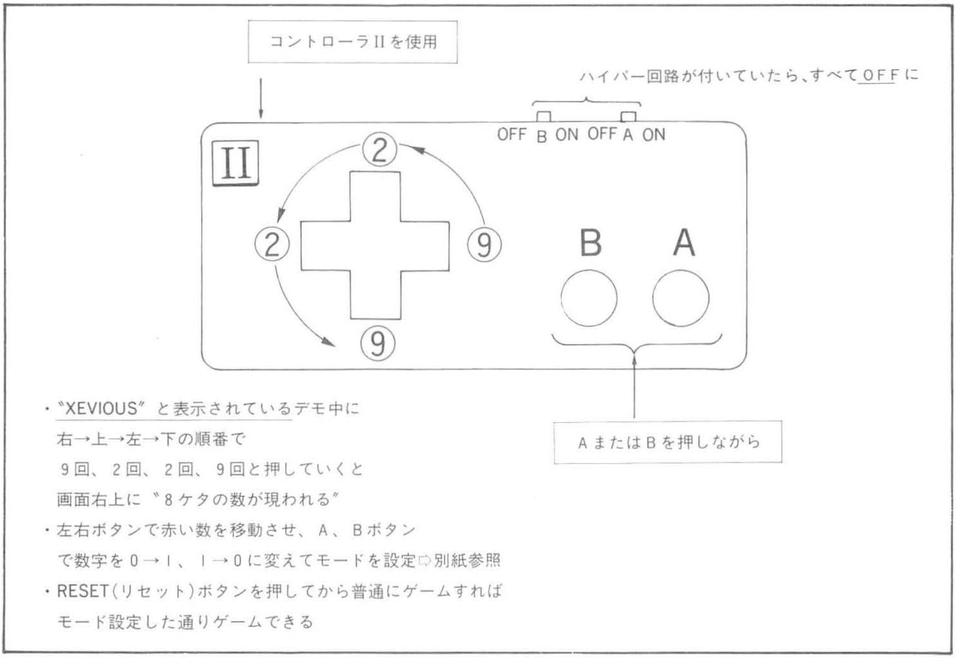
000	２万点で一機、	それ以後６万、	12万と６万点ごと
001	〃	〃　５万、	10万と５万点ごと
010	〃	〃　８万、	16万と８万点ごと
011	３万点で一機、	〃	〃　〃
100	〃	それ以後12万、	24万と12万点ごと
101	４万点で一機、	〃　８万、	16万と８万点ごと
110	〃	〃　10万、	20万と10万点ごと
111	５万点で一機、	それ以後15万、	30万と15万点ごと

■ ソルバルウがやられないモード設定

２通り
- 0 —— 当ったらやられる
- 1 —— 何に当ってもやられない

☆設定方式は、左右ボタンで変えたい数の場所へ赤い数を移動させる。
Ａボタンで０→１、１→０へと変える。ＯＫならリセットを押し、ゲームを始めれば良い。

〈表４〉ゼビウスのモード設定



〈図10〉 ゼビウスの隠しコマンドの実行方法



〈写真30〉 ゼビウス

ドルアーガの塔隠しコマンド

ナムコから出たファミコン初のロールプレイング・ゲームということで話題になっているゲームです。

60階まで上がってギルの恋人"カイ"を助けなければなりません。

各階では、次の階に行くためのカギと何かをすると出る宝の箱があります。宝の箱の出し方は各階でまったく異なり、その箱に入っている宝物を得ることでいろいろな機能を利用できるものです。

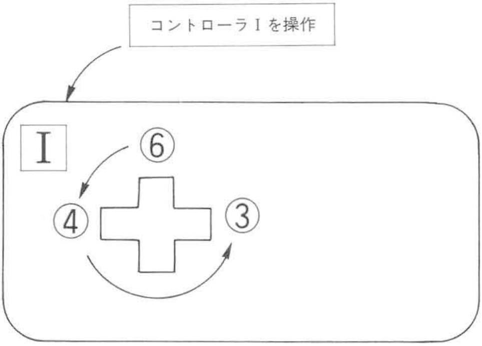
60階まで行けると、実はそこでゲームが終るのではなく、"裏ドルアーガ"という箱の出し方がまったく違うドルアーガが楽しめるという特典付きなんです。

ドルマーガの隠しコマンドを実行すると、60階まで行ったことがない人でも、裏ドルアーガを実行できてしまうのです。

コマンドの実行方法

コントローラⅠの方向キーを上、左、右の順番で6、4、3回押せばＯＫ！（図11）

今まで濃い灰色だった "DRUAGA" の文字が濃い緑色に変われば、裏ドルアーガになった印、電源を切らなければリセットを押しても裏ドルアーガのモードをぬけることはありません。



・方向キーを上、下
・方向キーを上、左、右の順で６、４、３回押せばＯＫ！
・裏ドルアーガになった印は〝ＤＲＵＡＧＡ〟の文字の色が
濃い緑色になることでわかります。

〈図11〉ドルアーガの隠しコマンドの実行方法



〈写真31〉裏ドルアーガのタイトル

保障はきかない！注意しよう

ハイパー改造、ビデオ出力改造など、どうしてもファミコンを開け改造しなければならないため、腕に自身のない人は改造をしない方が良いでしょう。もし、改造に失敗し、ファミコンが壊れたら保障期間内であったとしても保障は一切利きません。

〔タコ吸いなどに関して〕

タコ吸いでバックアップしたソフトやこの本に載っているプログラム、ハードなどはすべて自分個人が楽しむ範囲内でのみ使用し、それ以外は著者に無断で使用することはできません。

今回は、PC-8801mkIIを使用しましたが他のパソコン（FM-7やX1、MSXなど）でも入出力ポートを取り付ければ同様に楽しめます。ただ、ハードが異なるため、マシン語部はそのまま使用することはできません。BASIC部は一部の改造で使用できるでしょう。

次号では、FM-7などで動かそうと思ってます。

㊟ファミコンのハードなど、すべてこちらで解析したものです。よって任天堂への問い合わせは避けてください。

Part〈5〉

極秘テクニック

TENTACLES

ファミコンGAMEバックアップ法〝タコの吸い出し〞



ＭＳＸのカートリッジのバックアップはパート①でも紹介しましたが、今回は、ＭＳＸができるのならファミコンもということでファミコンのＲＯＭカセットをバックアップしてみることにしました。

ファミコンはPC88の周辺機器だ

ファミコンは、それ自体パソコンのようなモニタやBASICなどの言語を持っていないわけですから、ＭＳＸのようにバックアップすることはできません。

そこで、パソコンを使ってＲＯＭカセットの内容を吸い出しディスクやカセットにバックアップしようというものです。吸ったあとどうするのかって？ご心配なく！吸ったデータ（プログラム）は、今回製作した〝ＲＡＭカード〞に再び書込んで通常のＲＯＭカセットのように使用してしまうという画期的なものです。

つまり、パソコンは、プログラムやデータを一時貯えておく場所として利用するわけです。

もちろん、一時貯えておくだけではなく、ディスクやカセットにバックアップできるわけですから、友達の持っているＲＯＭカセットをバンバン・ディスクに入れてしまうことも可能なわけです。

また、〝ＲＡＭカード〞は電池でバックアップしていますから、ＲＯＭカセットと同じ感覚で、どこにでも持っていって楽しめるのです。

使用したパソコンは、手元にある関係でPC-8801mk IIを使用しました。ついにPC 88は、ファミコンを周辺機器にしたのです!!

タコの吸い出し基板

ＲＯＭカセットの内容を吸っちゃおうというのは良いのですが、データをやり取りするための出入口がありません。

一昔前のワンボード・マイコンでは、グラフィック機能を持っていなかったため、マイコンで何かを動かすことが一般的でした。そのためにデータをやり取りするための入出力ポートが当然のように付いていたのです。

パソコンが一般的になった今では、パソコンで何かを動かそうと考える人は少なくなり、入出力ポートは省略されてしまいました。残念なことです。

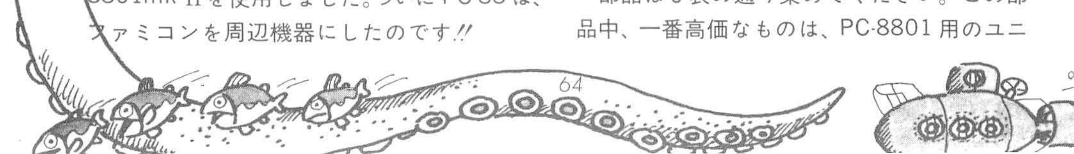
ＲＯＭカセットの内容を吸うという、まったく簡単な事のために入出力ポートを作らなければならないからです。

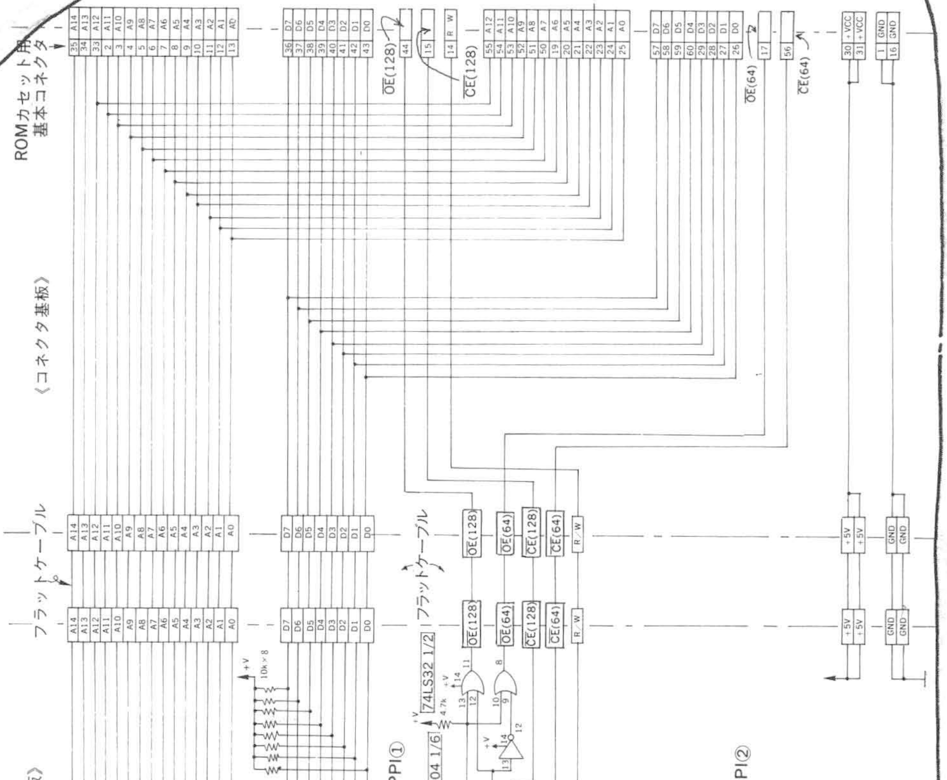
ＲＯＭカセットの内容を吸ったり、ＲＡＭカートリッジにデータを書き込んだりするための基板が、〝タコの吸い出し基板〞です。この基板は、8255というデータの入出力を行なうためのＩＣが２つ載り、１つは読み書きを行なう番地（アドレス）の指定と、データを送受するのに使い、もう１つは読み書きのコントロールを使います。（図12）

タコ吸い基板を作ろう

タコ吸い基板を作ることにしましょう。この基板は、PC 88で何かをコントロールしたいときにとても便利な物ですから１枚作っておくと良いでしょう。

部品は５表の通り集めてください。この部品中、一番高価なものは、PC-8801用のユニ





〈図12〉 ファミコン・タコの吸い出し回路図

バーサル基板です。サンハヤトのものは、ガラスエポキシを使い質は良いのですが高い。そう、すごいＩＣが載るのではないので安いものでＯＫです。また、60ピンのコネクタは一般的でないため、捜さなければならないかもしれません。

ＫＥＬ（ケル）のコネクタを扱っている店へ行ってみてください。注意として、このコネクタは、60ピンでもピン間隔が狭いタイプなので買う時、注意してください。（型番を指定すればＯＫ／）

・IC　8255AP–5		2
74LS 04		1
74LS 32		1
74LS 138		1
・ユニバーサル基板	PC-8801用ユニバーサル基板(サンハヤト)	1
	ICB-93（サンハヤト）	1
・ICソケット 40ピン		2
今回は使っていません	（14ピン）	2
	（16ピン）	1
・コンデンサ	電解 33μF 10V	1
・フラットケーブル	32芯	2m
・配線材	テフロン線など（RAMカードの分も入れて）	7～8m
・抵抗	4.7K（黄紫赤金）	1巻
・抵抗モジュール	10kΩ 8素子（ないときは10kΩ×8コ）	1
・60ピンコネクタ	KEL 4630-060-138（ピン間隔がせまいタイプなので注意）	1
・メッキ線	0.5ϕ	1巻

〈表５〉部品表〝タコ吸い出し基板〟

もしＲＯＭカセットがあるなら持って行き、合うか確認して買えば間違いありません。

部品がそろったら作ることにしましょう。PC 88 のユニバーサル基板は表と裏があります。表（部品面）を確認するために PC 88 の後部スロット（SLOT）１に基板を差し込んでみましょう。きっちり入ったときの上面が表で部品が付く側です。

基板の部品面が分かったら、まず8255が付く40ピンのソケットなどを取り付けます。

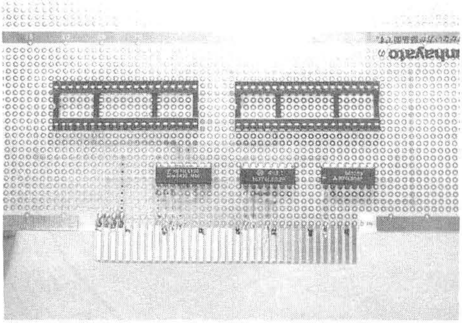
次にメッキ線などで電源ラインを配線し、テフロン線などで各端子とＩＣなどを配線します。チェックしてＯＫであれば、コネクタへ配線するためのフラット・ケーブルを取り付けます。

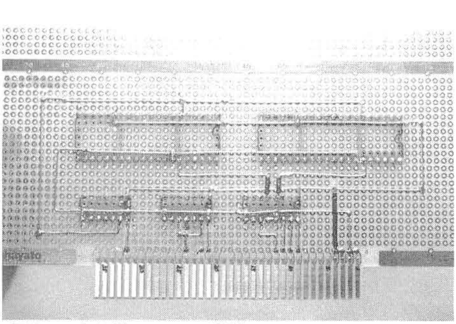
コネクタと基板は図14､15のように加工し、タコ吸い基板からのフラット・ケーブルを配線します。

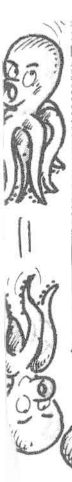
すべて配線し終わったら配線ミスがないか、もう一度チェックしてください。配線が完了した基板、コネクタの写真を載せておきますので参考にしてください。（写真34～37）

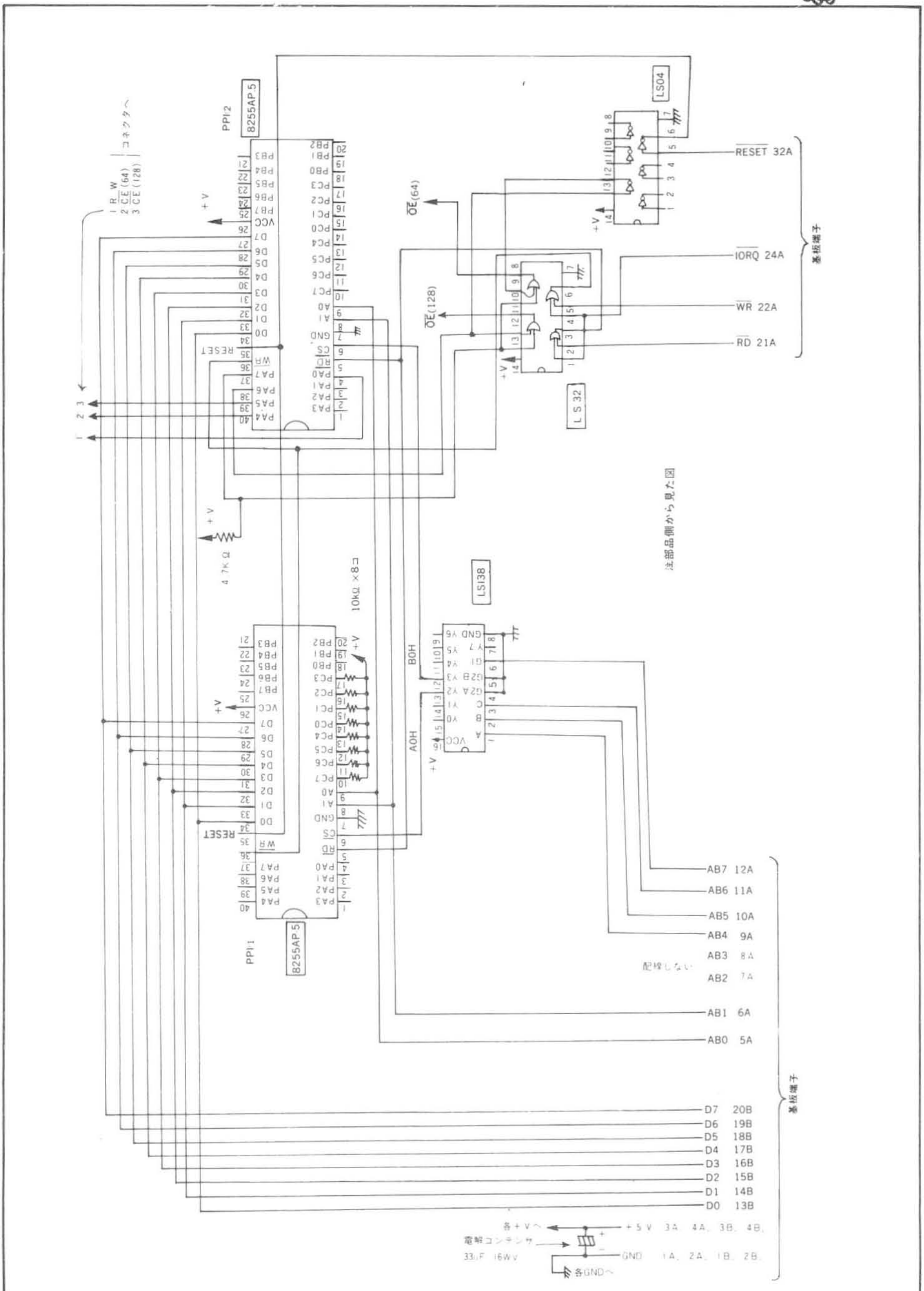
使ってみよう

この〝タコ吸い『テンタクルス』を PC 88 のスロット１（スロット２でも可）に差し込み、

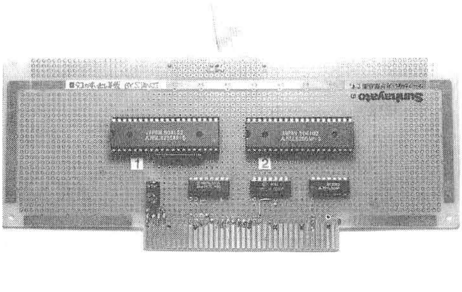

〈写真32〉


〈写真33〉うら側のジャンパ配線

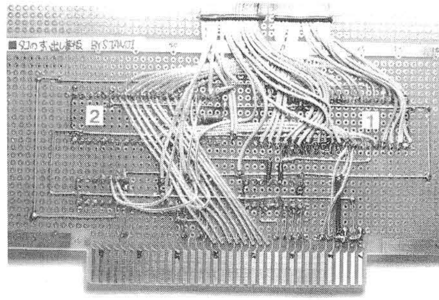




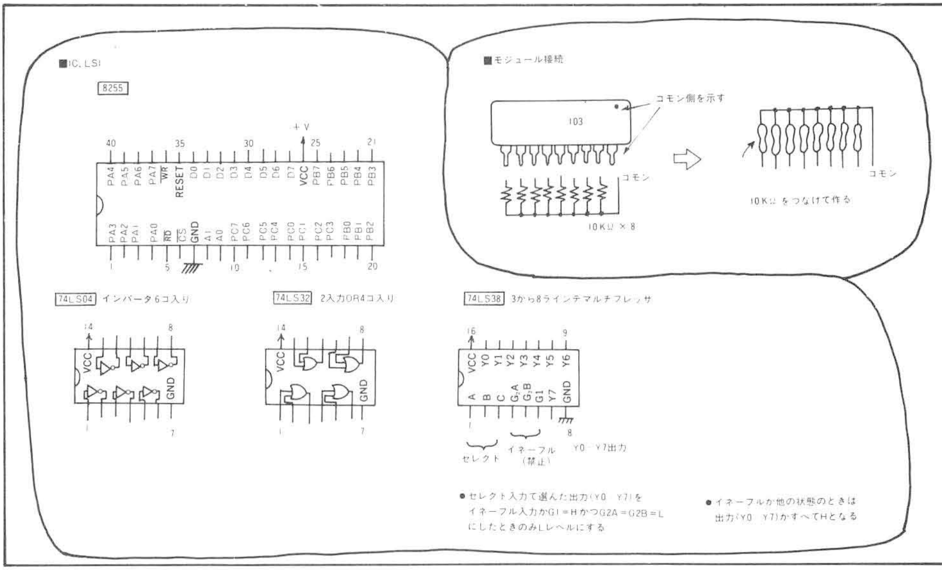
〈図13〉 電源などIC間の配線



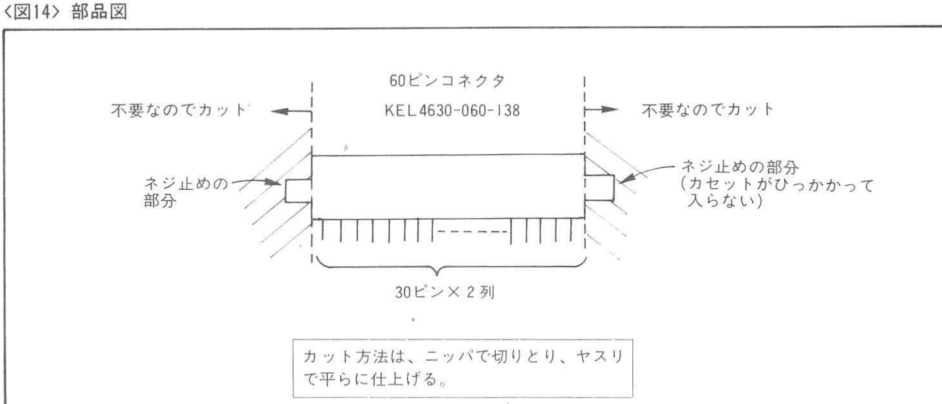
〈写真34〉 完成した〝タコの吸い出し基板部品面〟



〈写真35〉 完成した〝たこ吸い〟の〝ハンダ面の超アップ〟

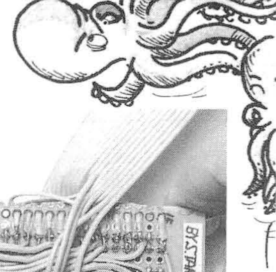


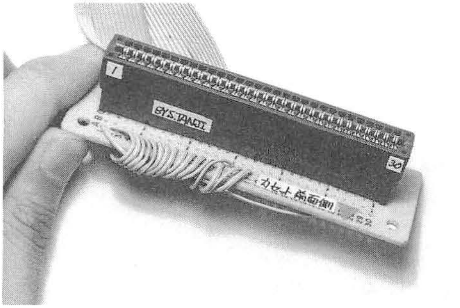
〈図14〉 部品図



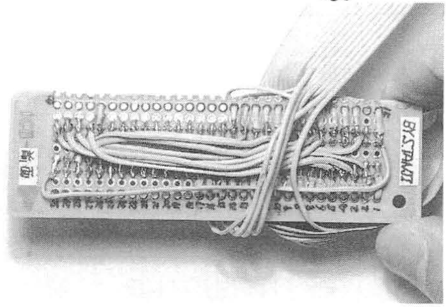
〈図15〉 コネクタの加工



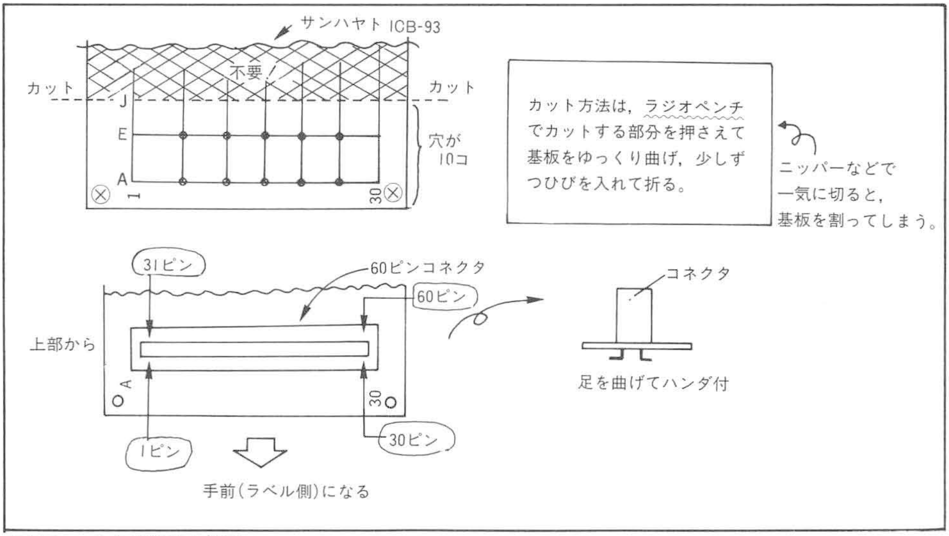




〈写真36〉 コネクタのアップ



〈写真37〉 コネクタのうら



〈図16〉 **コネクタ基板の加工**

電源をＯＮにしてみましょう。88が異常なく動作したら、コネクタのピン30、31(＋５Ｖ)とピン１、16（ＧＮＤ）間に＋５Ｖの電圧がちゃんとかかっているかをテスタで調べてください。

ＯＫ／ならコネクタにＲＯＭカセットを入れ（向きに注意）リスト１のプログラムを入れてください。

リスト１のプログラムは、ＲＯＭカセットまたは、ＲＡＭカードの内容を読んで88のメモリ（4000Ｈ～７ＦＦＦＨ番地）まで書き込むものです。メモリ・エリアはプログラムが入る最上部を使用しているのでCLEAR文などは不要です。チェック方法は表６の通り行ないます。



〈写真38〉 コネクタ"前面側"に合わせてROM カセットを入れる

〈リスト1〉

```
100 ST=&H0:LG=&H3F
110 RA=&H4000
120 PRO=&H1F:CHR=&H6F
130 ROM=PRO
140 OUT &HA3,&H89:OUT &HB3,&H89
150 OUT &HB0,ROM
160 '
170 FOR A=ST TO ST+LG
180 PRINT USING "&&00";RIGHT$("0"+HEX$(A),2)
190 FOR B=0 TO 255
200 OUT &HA0,B:OUT &HA1,A
210 POKE RA,INP(&HA2):RA=RA+1
220 NEXT:NEXT
230 BEEP
240 '==============================
250 ' Famicom ROM Read Program
260 '  Copyright 1985.8.13
270 '     by Saichi-Tanji
280 '==============================
```

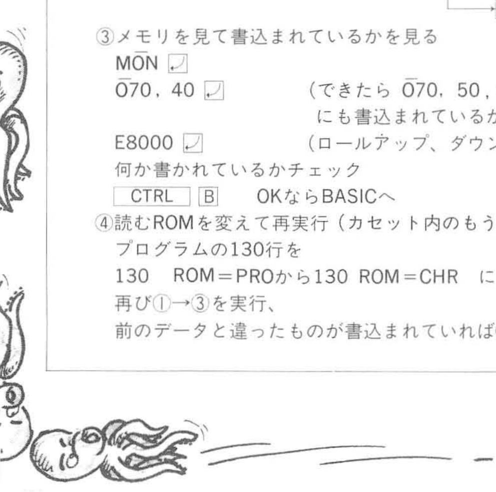
〈表6〉タコ吸いのチェック方法

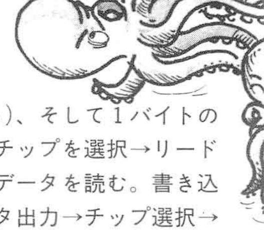
①メモリをクリアする
MON ↵ モニタに行く
F4000，7FFF，0 ↵　（4000～7FFF番地まで0（ゼロ）を入れる）
O70，40 ↵ ← 4000番地　｝4000番地から0が入っているか確かめる
E8000 ↵
CTRL B　（コントロールキーを押しながらBを押してBASICに戻る）

②プログラムをRUN
RUN ↵
プログラムが終るまで待つか、途中で止める → STOP キー

③メモリを見て書込まれているかを見る
MON ↵
O70，40 ↵　（できたら O70，50，O70，70などして5000番地や7000番地にも書込まれているかを見る）
E8000 ↵　（ロールアップ、ダウンキーで8000～83FFまでを見る）
何か書かれているかチェック
CTRL B　OKならBASICへ

④読むROMを変えて再実行（カセット内のもう1つのROMを読む。カセットの交換は不用）
プログラムの130行を
130　ROM＝PROから130 ROM＝CHR　にして
再び①→③を実行、
前のデータと違ったものが書込まれていればOK！





BASICでの使い方

タコ吸いのポート・アドレスは、PPI①（ROMカセットなどのアドレスとデータを受け持つ）が&HA0～、PPI②（ROMセレクトなどのコントロールを受け持つ）が&HB0～というように回路上で割り当てました。

もしこのポート・アドレスが使えないというのであればLS138のY_2、Y_3の出力をY_0～Y_7のどれかにします。ちなみに愛機のY_0(&H80）はツクモのファイナル・ウエポンV_3で使っていますからだめですし、Y_5(&HE0)からはシステムで使われてしまうので、結局のところY_1（&H90）～Y_4（&HC0）のどれかということになります。

各ポートは図17のようにROMカセットに接続されています。詳しい内容を図18に示しますが、ROM（RAM）を読み書きするためには、ハードで行なうタイミングなど、すべてソフトで行なわなければなりません。

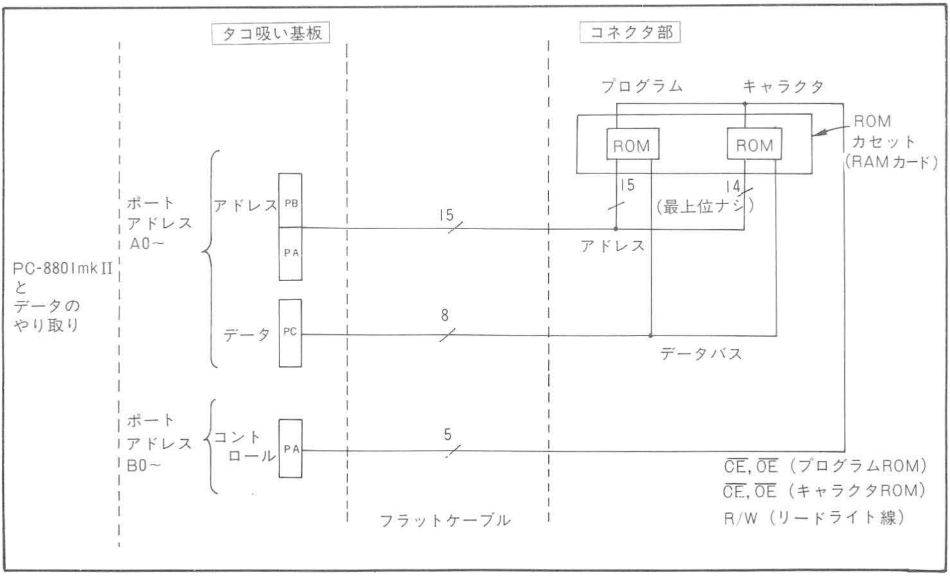
手順としては、ポートの使い方をまず設定し（コントロール・ワードをコントロール・ワード・レジスタに送る)、そして1バイトの読み書きは読むのならチップを選択→リード出力→アドレス出力→データを読む。書き込むのならアドレス、データ出力→チップ選択→ライト・パルス出力というように行ないます。（表7）

ファミコン・ダンプリスト

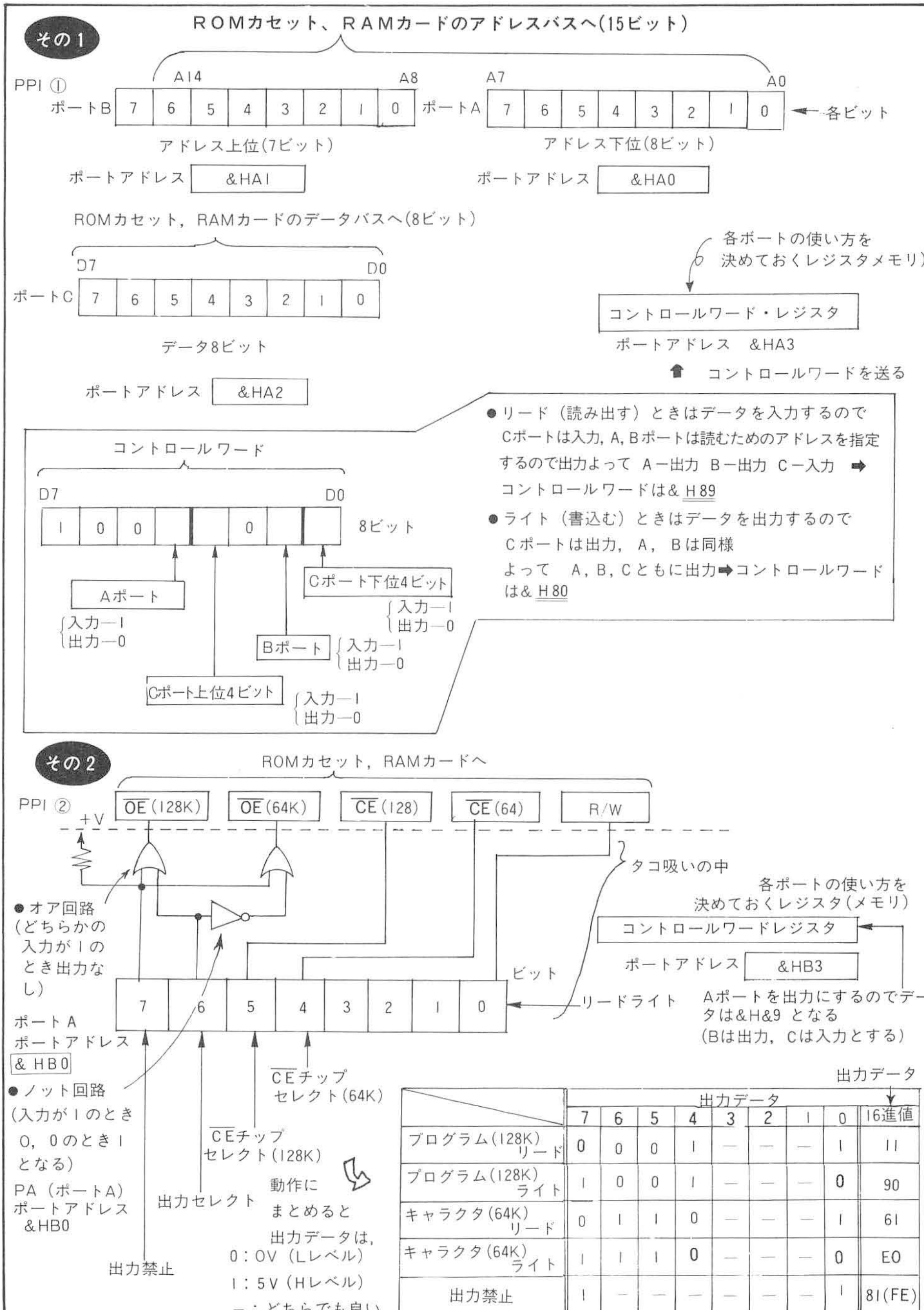
ファミコンのROMカセット（RAMカード）の内容をダンプ・リストして画面に表示するプログラムです。（リスト2）

このプログラムは、右側にキャラクタも表示してしまうため、普通は見ることのできない〝隠しメッセージ〞が見られるのです。写真39は、ゼビウスをダンプ・リストしたときのものです。プログラムの一番最後に〝隠しメッセージ〞なるものが入っているのがわかりますね。

プログラムの使い方は、プログラム入力後RUNすればOKです。見たいROMをプログラムなら〝P〞、キャラクタなら〝C〞のキーで選びます。するとROMの終わりまで表示します。

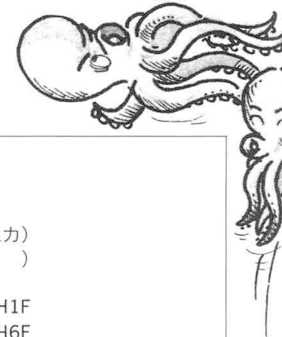


〈図17〉システム接続概要



	7	6	5	4	3	2	1	0	16進値
プログラム(128K) リード	0	0	0	1	—	—	—	1	11
プログラム(128K) ライト	1	0	0	1	—	—	—	0	90
キャラクタ(64K) リード	0	1	1	0	—	—	—	1	61
キャラクタ(64K) ライト	1	1	1	0	—	—	—	0	E0
出力禁止	1	—	—	—	—	—	—	1	81(FE)

〈図18〉 各ポートの使われ方



〈表7〉

読み出す

読み出す始めに1回行なえば良い
①コントロール，ワードを出力（各ポートの使い方を設定）
OUT ＆ HA3、＆ H89 ← PPI①（A、Bポート出力、Cポート入力）
OUT ＆ HB3，＆ H89 ← PPI②（A、B 〃 、 〃 ）

繰り返し
②読み出す ROM（RAM）を選ぶ
プログラム（128Kビットメモリ）のとき → OUT ＆HB0，＆H1F
キャラクタ（ 64Kビットメモリ）のとき → OUT ＆HB0，＆H6F
〔CE：チップセレクトをLレベル（チップを選択）、OE：アウトプットイネーブルをL（出力許可）にする〕

③読み出したいメモリのアドレスを出力
アドレスは 0000～&H7FFF（キャラクタは&H1FFF）までOK！

同じROM（RAM）の場合
OUT &HA0，（下位8ビットアドレス）
OUT &HA1，（上位8ビットアドレス）

例えば&H1250番地（16進値）を見たいときは
&H1250番地
上位 下位なので
OUT &HA0，&H50
OUT &HA1，&H12とする

④メモリからのデータを読む
（変数）＝ INP（&HA2）
0～255（16進で0～FF）までの値が入る

RAMカードを差し、WRITE（ライト：書込み）SWをONにしてください
（ROMカセットに対して行なっても書込めません、あしからず！）

書き込む

書込む始めに1回行なえば良い
①コントロールワードを出力（各ポートの使い方を設定）
OUT &HA3，&H80 ← PPI①（A、B、Cポート出力）
OUT &HB3，&H89 ← PPI②（A、Bポート出力、Cポート入力）

繰り返し
②書込むアドレスを出力する
アドレスは 0000～&H7FFF（キャラクタは&H1FFF）までOK！
OUT &HA0，（下位8ビットアドレス）
OUT &HA1，（上位8ビットアドレス）
方法は読み出すときと同じ

③書込むデータを出力する
データは 0～255（16進で0～&HFF）まで
OUT &HA2，（データ）
0～255 の範囲を越えるとエラーします。

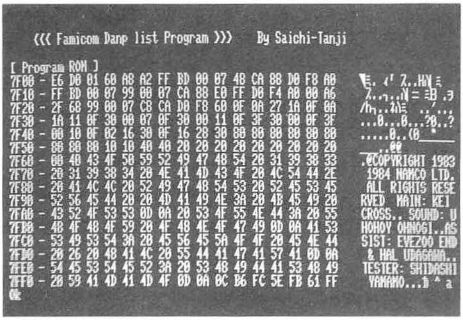
④データを書込む
書き込むには R/W（リードライト）線を H（ハイ、＋5V）から L（ロウレベル、0V）へ下げ、再びHに上げる必要があります。（Lレベルのとき書込みモードで、Hレベルのとき読出しモードになる）
・プログラム（128Kビット）の場合
OUT &HB0，&H91：OUT &HB0，&H90：OUT &HB0，&H91
・キャラクタ（64Kビット）の場合
OUT &HB0，&HE1：OUT &HB0，&HE0：OUT &HB0，&HE1

見たいアドレスを変えたいときは、行番号280のＳＴにスタート、ＥＤにエンド・アドレスを入れて実行すればＯＫです。

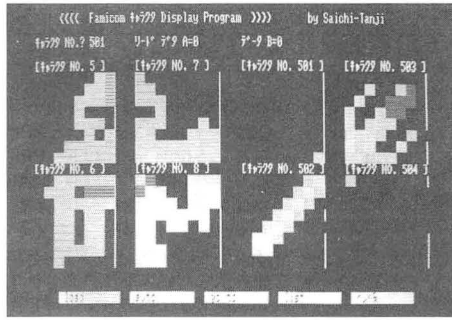
キャラクタディスプレイ・プログラム

ファミコンの楽しさの1つとして、キャラクタの動きのリアルさにあります。このキャラクタがどのようになっているのか見たくありませんか？このソフトは、ＲＯＭカセット（ＲＡＭカード）のキャラクタＲＯＭの内容を読み出し画面に表示するものです。（リスト3）

ファミコンのキャラクタは、前にも書きましたが８×８ドット構成ですが、色を付ける



〈写真39〉ROMの内容をダンプリストとして見れる〝ファミコンダンプリスト〟を使ってゼビウスを見たらプログラムの最後に隠しメッセージが入っていた。



〈写真40〉ROMカートリッジのキャラクタを表示させるソフト。ドルアーガのキャラクタでギルと剣を表示させた。

〈リスト2〉

```
100 CONSOLE 0,25:CLS:WIDTH 80,25
110 PRINT"     <<< Famicom Dump list Program >>>     By Saichi-Tanji"
120 CONSOLE 3,21
130 OUT &HA3,&H89:OUT &HB3,&H89
140 '
150 '   ROM Select
160 '
170 CHR=&H6F:LGC=&H1F ' Character ROM
180 PRO=&H1F:LGP=&H7F ' Program ROM
190 '
200 LOCATE 0,2:PRINT"ROM Select(P...Program  C...Character) ? ";:SE$=INPUT$(1)
210 LOCATE 0,2:IF SE$="p" OR SE$="P" THEN PRINT"[ Program";:ROM=PRO:LG=LGP ELSE
PRINT"[ Character";:ROM=CHR:LG=LGC
220 PRINT " ROM ]";SPC(60):PRINT:PRINT
230 '
240 OUT &HB0,ROM
250 '
260 '   Dump List Display
270 '
280 ST=0:ED=LG ' ST:Start address , ED:End address
290 FOR H=ST TO ED
300 FOR LL=0 TO 255 STEP 16
310 PRINT RIGHT$("000"+HEX$(H*256+LL),4);" - ";
320 FOR C=1 TO 2
330 FOR L=LL TO LL+15
340 OUT &HA0,L:OUT &HA1,H:D=INP(&HA2)
350 IF C=1 THEN PRINT RIGHT$("0"+HEX$(D),2);" ";:GOTO 370
360 IF D>31 THEN PRINT CHR$(D); ELSE PRINT".";
370 NEXT:PRINT"    ";:NEXT:PRINT:NEXT:NEXT
380 END
390 '
400 '===========================
410 '  Famicom Dump List Program
420 ' fmidmp2 Copyright 1985.8.30
430 '        by Saichi-Tanji
440 '===========================
```

ために同じく８×８ドットのパターンを用意し、２枚のパターンを重ねたときの点の有無で色を付けています。

ただ、このままですとキャラクタだけでは最大４色しか出せないため、キャラクタ毎などに別に色を指定するアトリビュートがあるものと思われます。今回は、その部分まで分からなかったため４色の表示となっています。(図19)

使い方は、ＲＵＮすると、表示したいキャラクタ番号を聞いてきますから１～512までの数を入力します。表示後、再びこの入力となりますが、現在表示したキャラクタ番号の次のものを表示させたいのであれば⏎（リターン）のキーのみの入力ですぐ表示してくれます。

また、１から順番に表示させたいときは、



〈リスト3〉

```
100 ON STOP GOSUB 510:STOP ON
110 CLS:CONSOLE ,,,1:WIDTH 80,25:GOSUB 370
120 OUT &HA3,&H89: ' port init
130 OUT &HB3,&H89
140 '
150 OUT &HB0,&H61: '64K bit ROM(Character) read
160 LOCATE 5,0:PRINT"<<<<  Famicom ｷｬﾗｸﾀ Display Program  >>>>        by Saichi-T
anji"
170 '------------------
180 '         ﾒｲﾝ ﾙｰﾁﾝ
190 '------------------
200 C1=2:C2=3:X1=0:Y1=4:X=X1:Y=Y1:' C1,C2 .. Disp Color(C3=C1+C2)
210 CH=1
220 COLOR 7:LOCATE 0,2:INPUT "ｷｬﾗｸﾀ NO.";CH$:IF CH$<>"" THEN CH=VAL(CH$)
230 IF CH=0 OR CH>512 THEN 220
240 ' IF CH>512 THEN COLOR 7:END
250 COLOR 7:LOCATE X,Y:PRINT"[ｷｬﾗｸﾀ NO.";CH;"]  ":CH=CH+1
260 Y=Y+1:IF Y>20 THEN Y=Y1:X=X+20:IF X>60 THEN X=X1
270 DD=16*(CH-1)
280 FOR DA=DD TO DD+7
290 H=INT(DA/256):L=DA-H*256:OUT &HA0,L:OUT &HA1,H:D=INP(&HA2):LOCATE 20,2:COLOR
 7:PRINT "ﾘｰﾄﾞ ﾃﾞﾀ A=";HEX$(D);" "
300 DE=DA+8:H=INT(DE/256):L=DE-H*256:OUT &HA0,L:OUT &HA1,H:E=INP(&HA2):LOCATE 40
,2:COLOR 7:PRINT "ﾃﾞｰﾀ B=";HEX$(E);" "
310 GOSUB 430
320 NEXT
330 GOTO 220
340 '--------------------
350 '         ﾃﾞﾀ ｾｯﾄ
360 '--------------------
370 FOR A=0 TO 7
380 D(A)=2^(7-A)
390 NEXT:RETURN
400 '--------------------
410 '        ﾃﾞｨｽﾌﾟﾚｲ
420 '--------------------
430 LOCATE X,Y
440 Y=Y+1:IF Y>21 THEN Y=Y1:X=X+20:IF X>60 THEN X=0
450 FOR A=0 TO 7
460 COLOR ABS((D AND D(A))<>0)*C1+ABS((E AND D(A))<>0)*C2
470 PRINT "■";
480 NEXT:PRINT
490 RETURN
500 '
510 COLOR 7:PRINT:PRINT CHR$(7);"ｴｰｯ! ﾄﾒﾁｬｳﾝﾃﾞｽｶ!!":END
520 '
530 '===================================
540 ' Famicom Character Display Progarm
550 '  Ver.1.2  Copyright 1985.8.12
560 '           by Saichi-Tanji
570 '===================================
```

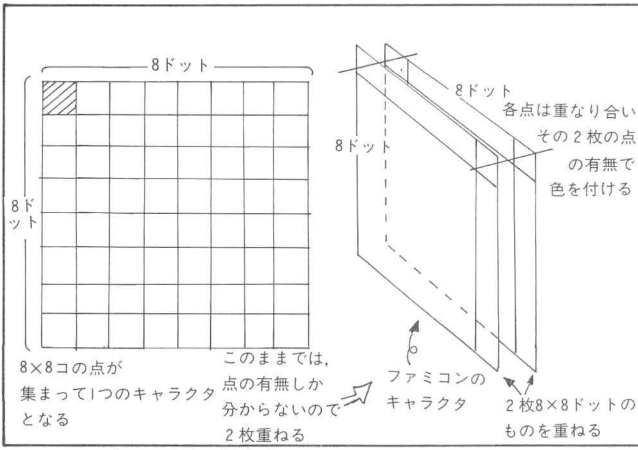
行番号220、230のプログラムの頭に'を入れREMに行にしたあと、240行に入っている'を取って実行すればキャラクタの最後まで見せてくれます。

㊟画面の右側のキャラクタを表示しているとき、色が出なかったり、キャラクタがくずれたりすることがあります。これは、プログラムが悪いのではなく、PC-8801（PC-8001からの）がもともと持っている機能の限界のためになってしまうのです。

キャラクタで色を付けたとき、横20カ所以上色を変えたりするとアトリビュート・エリアがいっぱいになり、キャラクタが表示されなかったり、色が出なかったりするのです。

キャラクタ・グラフィック画面表示

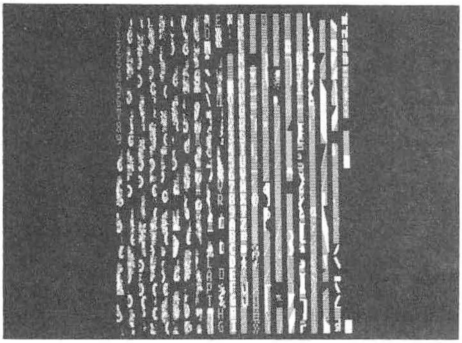
テキスト画面に表示していたのでは、実際に表示したときのキャラクタのイメージが出



点の有無		出力される色
2枚目	1枚目	
0	0	黒（透明）
0	1	色1
1	0	色2
1	1	色3

0のとき点なし
1のとき点あり

〈図19〉 ファミコンのキャラクタ



〈図41-A〉 Hiresキャラクタ表示画面（エキサイトバイク）
バイクの形が見える。

ませんから、高分解能画面にキャラクタを表示することにしました。（リスト4）

高分解能画面へのアクセスは、マシン語により行なっています。よって、バンクの切り換えをしてアクセスすることになるため、マシン語はＢＦ００Ｈ番地に置いています。

コネクタにＲＯＭカセット（ＲＡＭカード）を差し、ＲＵＮすればすべてのキャラクタを表示してくれます。表示される色は、行番号145の COLOR＝文の値を変えてください。

ＲＡＭカードの製作

ファミコン・バックアップの目玉商品〝ＲＡＭカード〟を製作することにしましょう。

ＲＡＭカードは、完全C-ＭＯＳスタティックＲＡＭ（ＴＣ5564ＰＬ-15）８Ｋバイトをプログラム用に４個(32Ｋバイト)、キャラクタ用に１個実装し、電池（３Ｖ）により長期間バックアップし続けます。

回路は割と簡単で、プログラム・エリアの32Ｋバイト分を８バイトごとのＲＡＭに振り分けるＴＣ40Ｈ138がＲＡＭの他にあるぐらいです。ただ、ほとんどのピンを並列に配線しなければならないので手間がかかると言えばかかります。（図20）

また、ＴＣ40Ｈ138というＩＣはハイスピードC-ＭＯＳといい、ＴＴＬ・ＩＣ74ＬＳ138とピン・コンパチブルのＩＣです。つまり、バックアップする関係で電気の食わないC-ＭＯＳを使用したわけです。使い方などは74ＬＳ138と同様です。

ＲＡＭカードのチェック

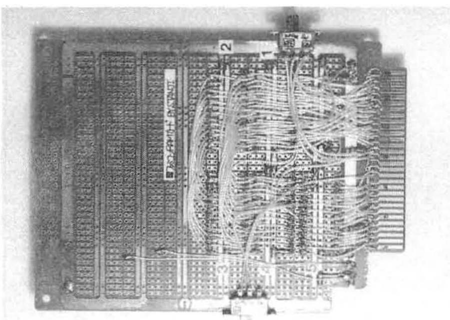
表８の通り部品を集めてください。ＲＡＭなどＩＣはC-ＭＯＳですので静電気に強くありません。配線が終わり、動作させるときまでアルミホイルなどで包んでおきましょう。

部品がそろったら、基板の加工をしましょう。60ピンのコネクタは見つかったものの、60ピンの基板を探すことはできませんでした。そこで、ピン数が72ピン（36×２）と多いサ

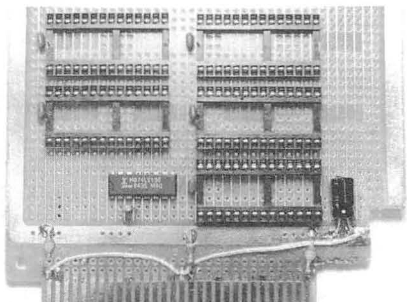


〈リスト4〉

```
100 CLS:CLEAR ,&HBF00:CONSOLE ,,0
110 DEF USR1=&HBF00:DEF USR2=&HBF05
120 OUT &HA3,&H89
130 OUT &HB3,&H89
140 '
145 COLOR=(4,7):COLOR=(6,1)
150 DD=0:D1=80*199:GOSUB 300
160 OUT &HB0,&H61: ' 64k read
170 FOR X=&HC000 TO &HC04F STEP 2
180 FOR Z=X TO X+D1 STEP 640
190 FOR AA=1 TO 2
200 FOR Y=Z TO Z+639 STEP 80
210 YY=65536!+Y:HH=INT(YY/256):LL=YY-HH*256
220 POKE &HBF1D,HH:POKE &HBF1C,LL
230 H=INT(DD/256):L=DD-H*256:DD=DD+1:IF DD>8192 THEN END
240 OUT &HA0,L:OUT &HA1,H
250 D=INP(&HA2):POKE &HBF1B,D
260 IF AA=1 THEN ZZ=USR1(0) ELSE ZZ=USR2(0)
270 NEXT:NEXT:NEXT:NEXT
280 END
290 '
300 IF PEEK(&HBF00)=&HC5 THEN RETURN
310 AD=&HBF00
320 READ D$:IF D$<>"ｵﾜﾘ" THEN POKE AD,VAL("&H"+D$):AD=AD+1:GOTO 320
330 RETURN
340 DATA C5,0E,5D,18,03,C5,0E,5E,F3
350 DATA E5,F5,3A,1B,BF,2A,1C,BF,ED
360 DATA 79,77,D3,5F,F1,E1,C1,FB,C9
370 DATA "ｵﾜﾘ"
380 '
400 '==============================
410 ' Famicom Character Hires Disp
420 '  hrsdsp v.1 Cpyright 1985.8
430 '       By Saichi-Tanji
440 '==============================
```



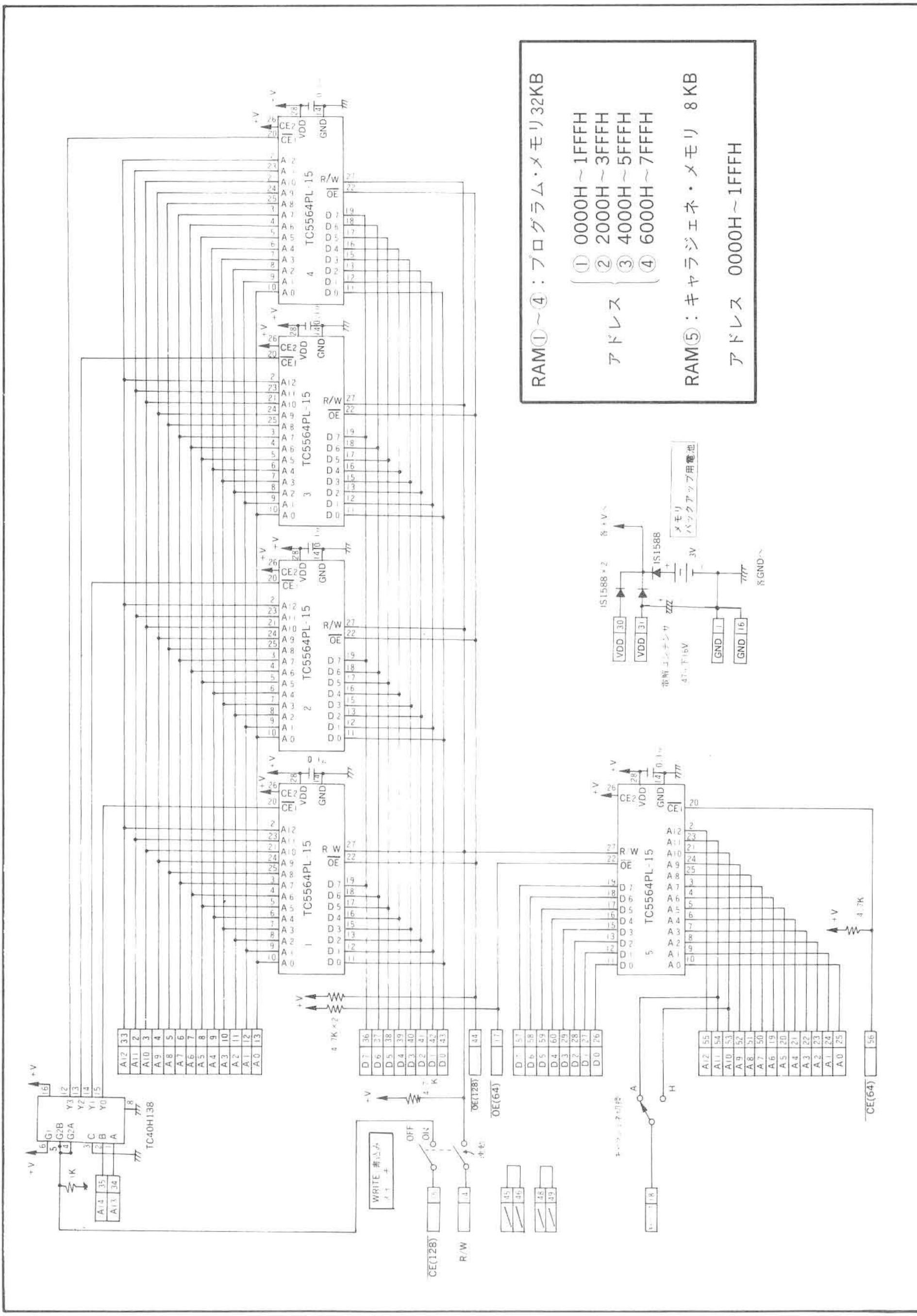
〈写真41-B〉"RAMカード"配線



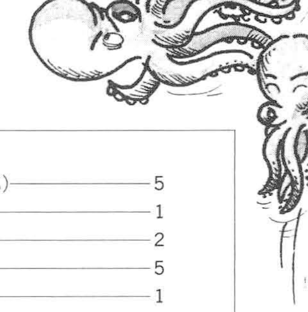
まずはソケットをつけてから

ンハトヤのCPU-109Fの端子を切り、それ用の基板とすることにしました。(図21)

基板がOKなら、ソケットをハンダ付けし、電源ラインを配線、続けて40H138から各ICの配線をします。チェックしてOKなら、アドレス線、データ線などすべてを配線します。配線時には、端子に端子NO。をマジックなどで入れておくとミスが減らせます。

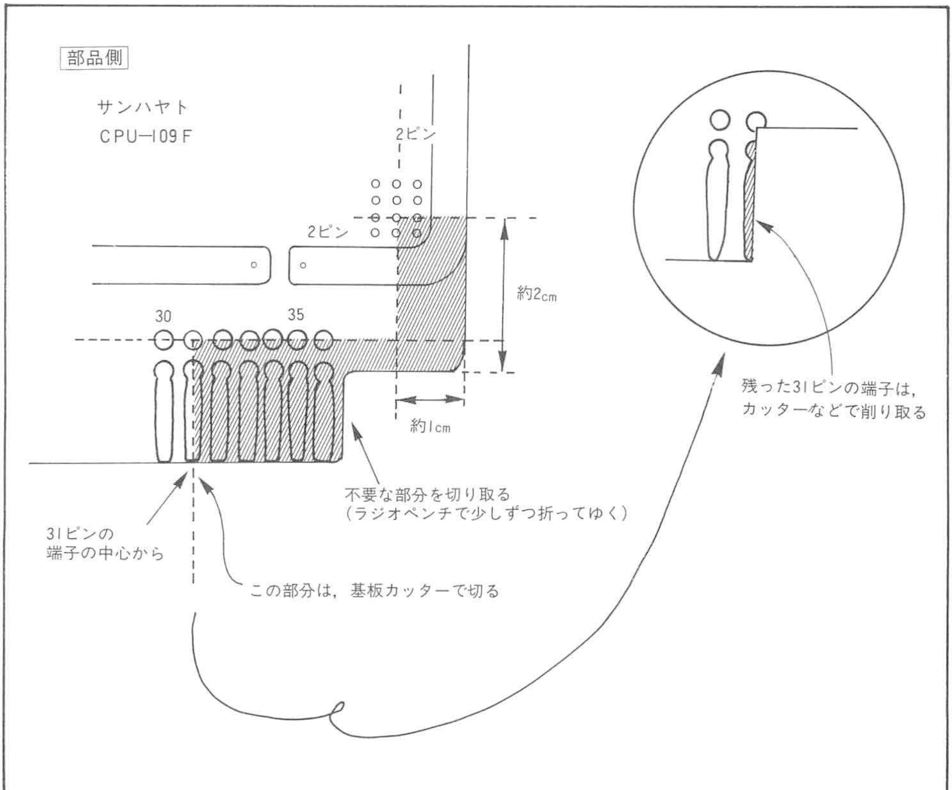


〈図20〉 ファミコンRAMカード回路図

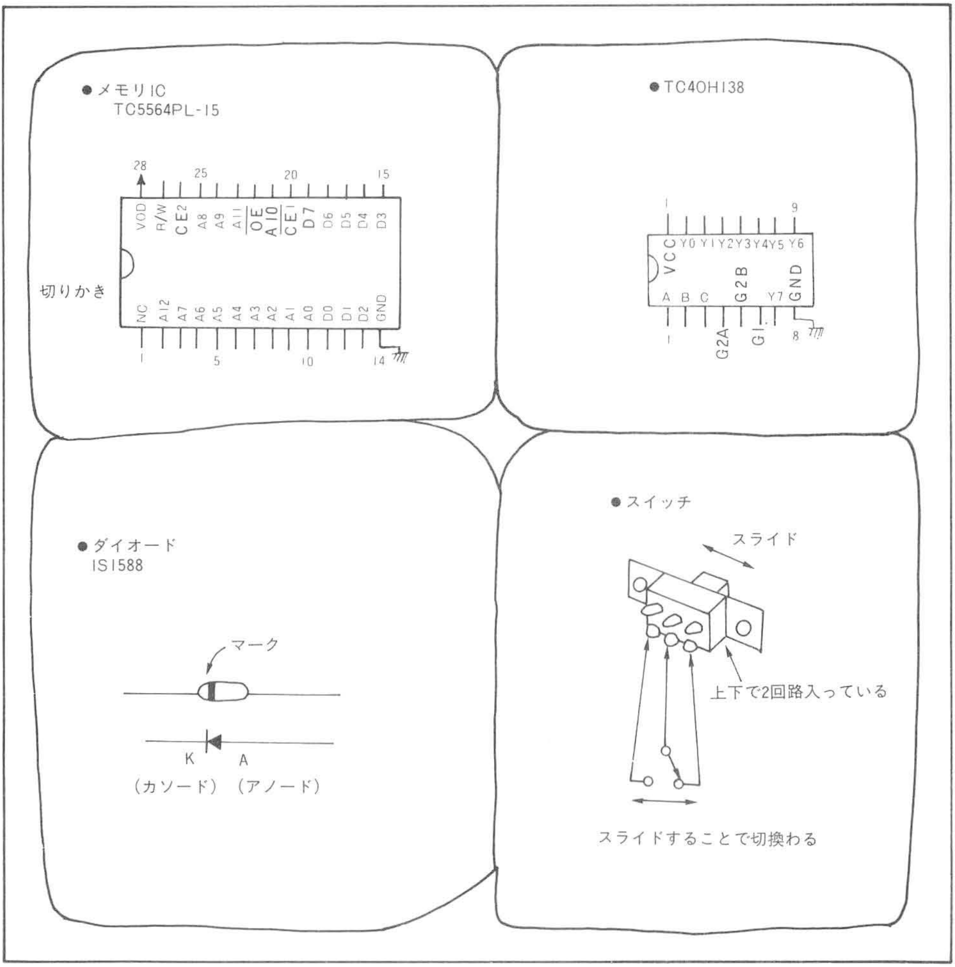


〈表 8〉 RAMカード部品表

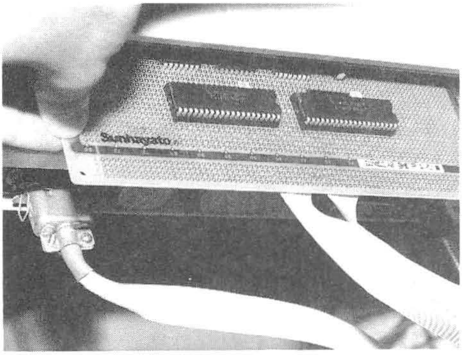
・IC	RAM　TC5564PL-15（8192ビット×8、アクセスタイム150ns）	5
	TC40H138	1
・ダイオード	IS1588	2
・ソケット	28ピン	5
	16ピン	1
・コンデンサ	0.1μF	5
	電解　47μF　16V	1
・抵抗	1kΩ　1/4W（茶黒赤金）	1
	4.7kΩ　1/4W（黄紫赤金）	4
・ユニバーサル基板	サンハヤト　CPU-109F(36ピン×2)-30ピン×2があればそれを買う	1
・スライドスイッチ	6P（2回路2接点）	2
・電池	単3	2
・電池ボックス	単3×2コ用	1
・配線材	テフロン線など——（タコ吸いなどと合わせて）	7～8m
・メッキ線	φ0.5	少々



〈図21〉 RAMカード基板加工図



〈図22〉 部品図





完成した〝RAMカード〟は写真41、44の通りです。

☆　　　　　　☆

完成したRAMカードをICに何も入ってない状態にしてコネクタに差し込みます。そして、88の電源を入れ、各ICの電源ピンにちゃんと電圧が加わっているかをテスタなどで確かめます。OKなら基板を抜き、40H138を入れ再びコネクタに差します。



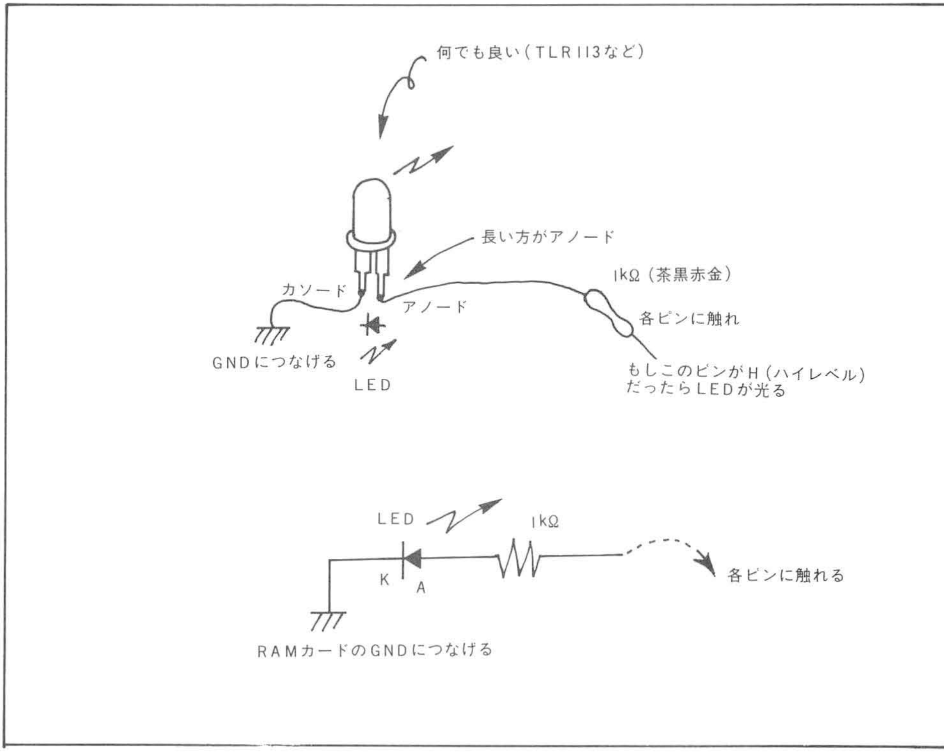
〈リスト5〉

```
100 OUT &HA3,&H80:OUT &HB3,&H89:PI=2
110 FOR A=1 TO 24
120 PRINT "pin no.";PI:PI=PI+1
130 IF PI=14 OR PI=26 THEN PI=PI+1
140 READ AH$,AL$,DT$,CM$
150 FOR B=1 TO 2
160 IF B=1 THEN OUT &HA1,0:OUT &HA0,0:OUT &HA2,0:OUT &HB0,0:GOTO 210
170 OUT &HA1,(VAL("&H"+AH$))
180 OUT &HA0,(VAL("&H"+AL$))
190 OUT &HA2,(VAL("&H"+DT$))
200 OUT &HB0,(VAL("&H"+CM$))
210 IF INKEY$<>"" THEN 240
220 FOR T=1 TO 5:NEXT T
230 NEXT B:GOTO 150
240 NEXT A:END
250 DATA 10,0,0,0
260 DATA  0,80,0,0
270 DATA  0,40,0,0
280 DATA  0,20,0,0
290 DATA  0,10,0,0
300 DATA  0,8,0,0
310 DATA  0,4,0,0
320 DATA  0,2,0,0
330 DATA  0,1,0,0
340 DATA  0,0,1,0
350 DATA  0,0,2,0
360 DATA  0,0,4,0
370 DATA  0,0,8,0
380 DATA  0,0,10,0
390 DATA  0,0,20,0
400 DATA  0,0,40,0
410 DATA  0,0,80,0
420 DATA  0,0,0,20
430 DATA  4,0,0,0
440 DATA  0,0,0,80
450 DATA  8,0,0,0
460 DATA  2,0,0,0
470 DATA  1,0,0,0
480 DATA  0,0,0,1
490 '=============================
500 ' Famicom RAMｶｰﾄﾞ Test Program
510 '      Copyright 1985.8.10
520 '        by Saichi-tanji
530 '=============================
```

ＲＡＭカードの各ピンの状態はチェックするために、ＬＥＤを使ったテスタ（図23）を作りましょう。このテスタは、ＬＥＤのカソード側をＲＡＭカードの㊀側に接続しておきアノード側の抵抗を調べたいピンに触れます。すると、ピンがＨレベルのときは、ＬＥＤが光って分かるというものです。

これをセットし、リスト５を実行してください。ＲＵＮさせると画面にチェックするピン№が表示されますから、そのピンにＬＥＤテスタの抵抗を付けてください。

もし、ちゃんと接続されていればチカチカ



〈図23〉 LEDテスタの製作

と点滅します。他のピンでは点滅もせず、同じピンではソケットが異なっても点滅することを確認してください。ＯＫだったらスペース・バーなどを押して次のピンへとチェックしていきます。（$\overline{\text{C E}}$ ：ピン22はこのプログラムでは、ソケット１しか正しく動きません）

書き込んでみよう

一応ＯＫだったらＩＣの向きに注意してソケットに差し、バックアップ用の電池も入れます。Write（ライト）スイッチをＯＮにしてからリスト６を入れ実行してみてください。

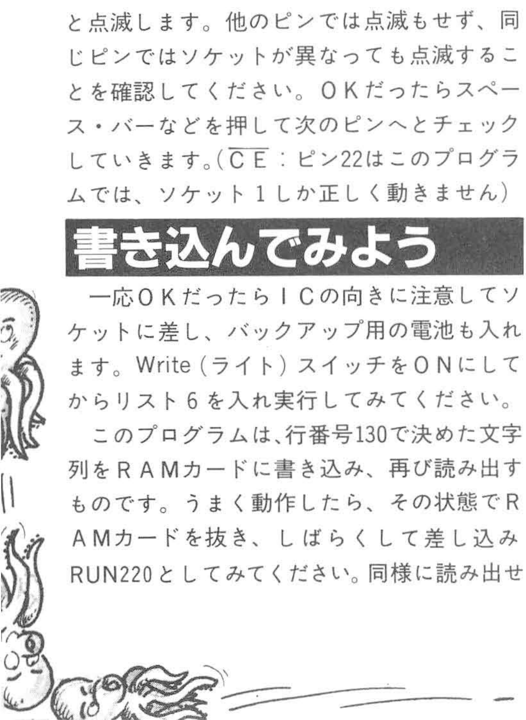
このプログラムは、行番号130で決めた文字列をＲＡＭカードに書き込み、再び読み出すものです。うまく動作したら、その状態でＲＡＭカードを抜き、しばらくして差し込みRUN220としてみてください。同様に読み出せれば、バックアップもＯＫです。

BASICで使うには

BASICでコントロールするための方法は前に説明しましたが、実際のプログラムにするとどうなるでしょう。

リスト７では、88のＲＡＭエリア＆ＨＥ000～Ｅ５ＦＦまでを使ってＲＯＭカセットの内容を読んだり、逆にそのエリアの内容を書込んだりしています。

ＲＵＮ 100で88に読み込み、ＲＵＮ 300とすると88のメモリ内容をＲＡＭカードに書き込みます。これを発展させればＲＯＭカセットの内容を吸っちゃうプログラムができるわけです。





〈リスト6〉

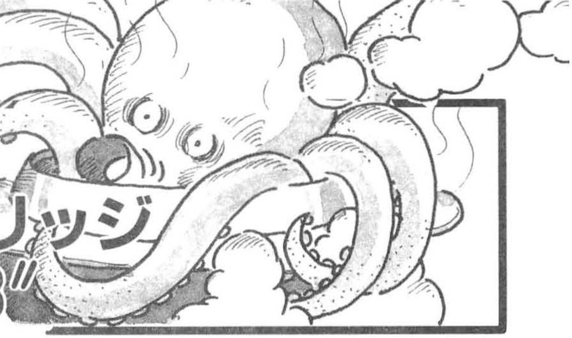
```
100 OUT &HA3,&H80:OUT &HB3,&H89:AD=0
110 '
120 PRINT"<RAM Write>"
130 D$="Famicom RAM ｶｰﾄﾞ TEST!!  ﾁｬﾝﾄ ｶｷｺﾏﾚﾃｲﾏｽｶ ?"
140 LG=LEN(D$)
150 FOR F=1 TO LG
160 D=ASC(MID$(D$,F,1)):PRINT CHR$(D);
170 OUT &HA0,AD:OUT &HA1,0:AD=AD+1:OUT &HA2,D
180 OUT &HB0,&HE1
190 OUT &HB0,&HE0:OUT &HB0,&HE1
200 NEXT:PRINT
210 '
220 PRINT:PRINT"<RAM Read>":AD=0
230 OUT &HA3,&H89:OUT &HB0,&H61
240 FOR A=0 TO LG-1
250 OUT &HA0,A:OUT &HA1,0:PRINT CHR$(INP(&HA2));
260 NEXT
270 END
280 '
290 '======================
300 ' RAM ｶｰﾄﾞ TEST Program
310 '  Copyright 1985.8.30
320 '    by Saichi-Tanji
330 '======================
```

〈リスト7〉

```
100 '----------------
110 '       Read
120 '----------------
130 CLEAR ,&HE000
140 PRINT"ROM ｶｾｯﾄ ｦ ｲﾚﾃｸﾀﾞｻｲ!! Okﾅﾗ ﾅﾆｶ ｷｰｦ ｵｼﾃｸﾀﾞｻｲ ";:Z$=INPUT$(1)
150 OUT &HA3,&H89:OUT &HB3,&H89
160 AD=0:OUT &HB0,&H61
170 FOR A=&HE000 TO &HE5FF
180 H=INT(AD/256):L=AD-H*256:AD=AD+1
190 OUT &HA0,L:OUT &HA1,H
200 D=INP(&HA2):PRINT USING"&  & - &&";HEX$(A),RIGHT$("0"+HEX$(D),2)
210 POKE A,D
220 NEXT
230 END
300 '----------------
310 '       Write
320 '----------------
330 CLEAR ,&HE000
340 PRINT"RAM ｶｰﾄﾞ ｦ ｲﾚﾃｸﾀﾞｻｲ!! Okﾅﾗ ﾅﾆｶ ｷｰｦ ｵｼﾃｸﾀﾞｻｲ ";:Z$=INPUT$(1)
350 OUT &HA3,&H80:OUT &HB3,&H89
360 AD=0:OUT &HB0,&HE1
370 FOR A=&HE000 TO &HE5FF
380 H=INT(AD/256):L=AD-H*256:AD=AD+1
390 OUT &HA0,L:OUT &HA1,H
400 D=PEEK(A):OUT &HA2,D:OUT &HB0,&HE1
410 OUT &HB0,&HE0:OUT &HB0,&HE1
420 PRINT USING"&  & - &&";HEX$(A),RIGHT$("0"+HEX$(D),2)
430 NEXT
440 '
450 '=========================
460 '  RAMｶｰﾄﾞ Read/Write Test
470 '   Copyright   1985.8.30
480 '     By Saichi-Tanji
490 '=========================
```

Part〈6〉ファミコンGAMEカートリッジ

バックアップ・プログラム "tako 88"



ファミコンのROMカセットの内容を読み込んで、ディスクやカセットに保存でき、さらにRAMカードを使うことで、バックアップしたゲームが楽しめるという〝ファミコン・バックアップ・プログラム〟を紹介しましょう。(リスト8)

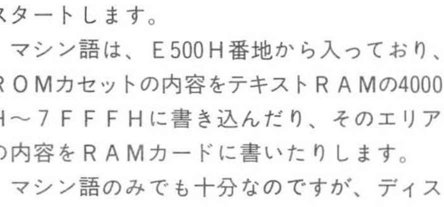
タコの吸い出し "tako 88"

プログラムは、BASIC＋マシン語でできており、RUNするだけでマシン語を書き込みスタートします。

マシン語は、E500H番地から入っており、ROMカセットの内容をテキストRAMの4000H〜7FFFHに書き込んだり、そのエリアの内容をRAMカードに書いたりします。

マシン語のみでも十分なのですが、ディスクのアクセスや操作性をアップさせるために、BASICとリンクしました。

メモリは、テキスト・エリアの半分である16Kバイトをバッファとして、利用していますが、ROMカセットのプログラムの長さは32Kバイトであるため、2回のアクセスを必要とし、さらにキャラクタのデータが8Kバイトあるため、1つのROMカセットに対して3つのファイルが作られます。ただ操作は、大まかなファイル名の入力ですべてが自動で行なわれますので特に気にしなくてもOKです。



〈写真42〉ファミコンにRAMカードを入れて実行

〔RAMカード使用時の注意〕

RAMカード(ROMカセット)のぬき差しは、特に電源のON/OFFを気にしなくてOKです。ただ、ファミコン側に差すときは、電源を切ってから差し込んでください。

RAMカードにデータを書き込む時は、必ずwrite(ライト)SwをONにし、ファミコンに入れる時は必ずOFFとしてください。これは、ROMカセットで使用していないピンを$\overline{R}$/Wなどの信号に割り当てていますがファミコン側でこれらのピンを使っているため、OFFにして切り離さないと、せっかく書き込んだものが壊されてしまいます。

また、キャラジェネ・スイッチはROMカセットにより、正しい絵が出る側に切り換えて使ってください。画面を見ながら、切り換え、おかしければリセットSwを押しもう一度というように行なってください。

使ってみよう

RUNするとメッセージが表示されますので実行したい内容の番号(1〜6)を入れてください。(⏎キーは不要)

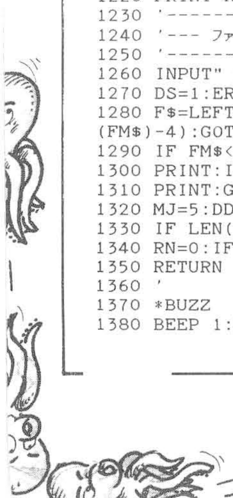
②Auto Save(オートセーブ)モード

このモードは、ROMカセットの内容をディスクやカセットにオートマチックで入れるモードです。ROMカセット(RAMカード)

〈リスト 8〉

```
100 CLEAR ,&HE500:WIDTH 80,25:IF PEEK(&HE5DE)<>&HF1 THEN GOSUB 1760
110 DIM SF$(10),MS$(10),DV$(10)
120 ON STOP GOSUB 1650:STOP ON:ON ERROR GOTO 1660
130 DEF USR0=&HE500 ' port init
140 DEF USR1=&HE589 ' program read
150 DEF USR2=&HE58D ' program write
160 DEF USR3=&HE595 ' character read
170 DEF USR4=&HE599 ' character write
180 MAD=&HE5A1:SF$(1)="1.pro":SF$(2)="2.pro":SF$(3)=".chr"
190 DV$(1)=" Disk ":DV$(2)=" Cassette "
200 MS$(1)="Auto LOAD":MS$(2)="Auto SAVE":MS$(3)="Back Up"
210 MS$(4)="Manual Read":MS$(5)="Manual Write":MS$(6)="Quit"
220 MN=&HE66C:POKE MN,&HC3:POKE MN+1,&HB0:POKE MN+2,&HE5
230 '
240 PRINT STRING$(80,"-");
250 PRINT STRING$(10,CHR$(&H87));"  Famicom Back Up Program  ";
260 PRINT STRING$(10,CHR$(&H87)),"By Saichi-Tanji"
270 PRINT STRING$(80,"-");:CONSOLE 3,21
280 '
290 ZZ=USR0(0) ' port init
300 '
310 GOSUB 1480:FOR T=1 TO 500:NEXT
320 FOR T=1 TO 30:PRINT:NEXT:LOCATE 0,3
330 '
340 ON KY GOSUB 520,370,680,850,1180,1650
350 IF ER<>1 THEN PRINT:PRINT"Complete !!":GOSUB *BUZZ:FOR T=1 TO 300:NEXT
360 GOTO 310
370 '------------------
380 '---- ｵｰﾄ ｾｰﾌﾞ ----
390 '------------------
400 PRINT"[ Auto SAVE ]":PRINT:COLOR 2:PRINT" ROM ｶｾｯﾄ ｦ ｲﾚﾃｸﾀﾞｻｲ !!":COLOR 0
410 GOSUB 1240:IF ER=1 THEN RETURN
420 CT=1:FOR A=1 TO 2
430 FOR B=1 TO 2
440 PRINT ".. ROM ｶｾｯﾄ ﾘｰﾄﾞ! ",
450 IF B=1 THEN POKE MAD,0 ELSE POKE MAD,&H40
460 IF A=1 THEN ZZ=USR1(0):LG=&H4000:CE$="7fff" ELSE ZZ=USR3(0):LG=&H2000:CE$="5fff"
470 FI$=FM$+SF$(CT):PRINT DV$(DS);"ｾｰﾌﾞ.. ";FI$
480 IF DS=1 THEN BSAVE FI$,&H4000,LG:GOTO 500
490 D$="w"+FI$+",4000,"+CE$+CHR$(13)+CHR$(2):GOSUB *MCOM
500 IF A=2 AND B=1 THEN RETURN
510 CT=CT+1:NEXT:NEXT:END
520 '------------------
530 '---- ｵｰﾄ ﾛｰﾄﾞ ----
540 '------------------
550 PRINT"[ Auto LOAD ]":PRINT:COLOR 2:PRINT" RAM ｶｰﾄﾞｦ ｲﾚ  Write SW ｦ ON ﾆ ｼﾃｸﾀﾞｻｲ !!":COLOR 0
560 GOSUB 1240:IF ER=1 THEN RETURN
570 CT=1:FOR A=1 TO 2
580 FOR B=1 TO 2
590 FI$="":IF RN<>1 THEN FI$=FM$+SF$(CT)
600 PRINT ".. ";DV$(DS);" ﾛｰﾄﾞ ";FI$,
610 IF DS=1 THEN BLOAD FI$,&H4000:GOTO 630
620 D$="r"+FI$+CHR$(13)+CHR$(2):GOSUB *MCOM
630 IF B=1 THEN POKE MAD,0 ELSE POKE MAD,&H40
640 IF A=1 THEN ZZ=USR2(0) ELSE ZZ=USR4(0)
650 PRINT".. RAM ｶｰﾄﾞ Write !!"
660 IF A=2 AND B=1 THEN RETURN
670 CT=CT+1:NEXT:NEXT:END
680 '--------------------------
690 '-- ROM ｶｾｯﾄ to RAM ｶｰﾄﾞ --
700 '--------------------------
710 PRINT"[ ROMｶｾｯﾄ to RAM ｶｰﾄﾞ Back up ]":PRINT
720 COLOR 2:PRINT"  RAM ｶｰﾄﾞﾊ Write Sw ｦ ON ﾆｼﾃｵｸｺﾄ !!":COLOR 0
730 FOR A=1 TO 2
740 IF A=1 THEN PRINT".. Program" ELSE PRINT".. Character"
```

```
750 FOR B=1 TO 2
760 IF B=1 THEN POKE MAD,0 ELSE POKE MAD,&H40
770 COLOR 4:PRINT"ROM ｶｾｯﾄ";:COLOR 0:PRINT" ｦ ｻｼｺﾝﾃﾞ ｸﾀﾞｻｲ .. ok!ﾅﾗ ﾅﾆｶ ｷｰｦ ｵｼﾃ!
!";:V=CSRLIN
780 GOSUB *BUZZ:Z$=INPUT$(1):LOCATE 23,V:PRINT SPC(40):GOSUB *BUZZ:PRINT
790 IF  A=1 THEN ZZ=USR1(0) ELSE ZZ=USR3(0)
800 COLOR 6:PRINT"RAM ｶｰﾄﾞ";:COLOR 0:PRINT" ｦ ｻｼｺﾝﾃﾞ ｸﾀﾞｻｲ .. ok!ﾅﾗ ﾅﾆｶ ｷｰｦ ｵｼﾃ!
!";:V=CSRLIN
810 GOSUB *BUZZ:Z$=INPUT$(1):LOCATE 23,V:PRINT SPC(40):GOSUB *BUZZ:PRINT
820 IF  A=1 THEN ZZ=USR2(0) ELSE ZZ=USR4(0)
830 IF A=2 THEN RETURN
840 NEXT:PRINT:PRINT:NEXT:END
850 '--------------------
860 '--- Manual Read -----
870 '--------------------
880 WR=0:DV$(0)="ROMｶｾｯﾄ ":PRINT"[ Manual Read ]":PRINT
890 PRINT" ﾒﾃﾞｨｱ ﾊ (R...";DV$(0);"/ D...Disk or ｶｾｯﾄ) ?";:K$=INPUT$(1)
900 D=INSTR("rRdD",K$):IF D=0 THEN ER=1:RETURN
910 DS=INT((D-1)/2):PRINT DV$(DS);:IF DS=1 THEN PRINT "or";DV$(2);
920 PRINT"ﾓｰﾄﾞ":IF DS>=1 THEN GOSUB 1240:IF ER=1 THEN RETURN
930 PRINT:PRINT"     << ﾅｲﾖｳ >>"
940 PRINT" (1) .. Program ｾﾞﾝﾊﾝ(0000-3FFF)"
950 PRINT" (2) .. Program ｺｳﾊﾝ (4000-7FFF)"
960 PRINT" (3) .. Character":PRINT
970 PRINT"  Serect(1-3) ?";:KY$=INPUT$(1):PRINT KY$:GOSUB *BUZZ
980 ER=0:IF ASC(KY$)<&H30 THEN ER=1:RETURN ELSE KY=VAL(KY$)
990 IF KY<1 OR KY>3 THEN 970
1000 FI$="":IF RN<>1 THEN FI$=FM$+SF$(KY)
1010 IF DS>=1 THEN PRINT"  ﾌｧｲﾙ .. ";FI$:PRINT
1020 POKE MAD,0:IF KY=2 THEN POKE MAD,&H40
1030 IF WR=0 THEN 1120
1040 '
1050 ON DS+1 GOTO 1060,1070,1090
1060 IF KY<=2 THEN ZZ=USR2(0):RETURN ELSE ZZ=USR4(0):RETURN
1070 IF KY<=2 THEN LG=&H4000 ELSE LG=&H2000
1080 BSAVE FI$,&H4000,LG:RETURN
1090 IF KY<=2 THEN LG$="7fff" ELSE LG$="5fff"
1100 D$="w"+FI$+",4000,"+LG$+CHR$(13)+CHR$(2):GOSUB *MCOM:RETURN
1110 '
1120 ON DS+1 GOTO 1130,1140,1150
1130 IF KY<=2 THEN ZZ=USR1(0):GOTO 1160 ELSE ZZ=USR3(0):GOTO 1160
1140 BLOAD FI$,&H4000:GOTO 1160
1150 D$="r"+FI$+CHR$(13)+CHR$(2):GOSUB *MCOM
1160 PRINT:PRINT"... ｶｷｺﾝﾀﾞﾒﾓﾘ ｱﾄﾞﾚｽ(4000H-7FFFH)":GOSUB *BUZZ:FOR T=1 TO 800:NE
XT
1170 RETURN
1180 '--------------------
1190 '--- Manual Write -----
1200 '---------------------
1210 DV$(0)="RAM ｶｰﾄﾞ ":WR=1:PRINT"[ Manual Write ]":PRINT
1220 PRINT"Read RAM ｱﾄﾞﾚｽ(4000H-7FFFH)":PRINT:GOTO 890
1230 '--------------------
1240 '--- ﾌｧｲﾙﾈｰﾑ input ---
1250 '--------------------
1260 INPUT" ﾌｧｲﾙﾈｰﾑ";FM$
1270 DS=1:ER=0:IF FM$="" THEN ER=1:RETURN
1280 F$=LEFT$(FM$,4):IF F$="cas:" OR F$="CAS:" THEN MJ=5:DS=2:FM$=RIGHT$(FM$,LEN
(FM$)-4):GOTO 1330
1290 IF FM$<>"files" AND FM$<>"files 2" THEN 1320
1300 PRINT:IF VAL(RIGHT$(FM$,1))<1 THEN FILES ELSE FILES 2
1310 PRINT:GOTO 1260
1320 MJ=5:DD$=LEFT$(FM$,2):IF DD$="1:" OR DD$="2:" THEN MJ=7
1330 IF LEN(FM$)>MJ THEN GOSUB *BUZZ:PRINT"<5ﾓｼﾞﾏﾃﾞﾃﾞｽ!!>":GOTO 1240
1340 RN=0:IF DS=2 AND FM$="" THEN RN=1
1350 RETURN
1360 '
1370 *BUZZ
1380 BEEP 1:FOR T=1 TO 100:NEXT:BEEP 0:RETURN
```





```
1390 '------------------
1400 '---- ﾓﾆﾀ ｺﾏﾝﾄﾞ ---
1410 '------------------
1420 *MCOM
1430 L=LEN(D$)
1440 FOR AA=1 TO L:POKE &HEFD8+AA,ASC(MID$(D$,AA,1)):NEXT
1450 POKE &HEFCD,L-1:POKE &HEFCE,31
1460 MON:RETURN
1470 '---------------------
1480 '----- Menu disp -----
1490 '---------------------
1500 CLS:H=22:V=3:HH=30:LOCATE H,V:PRINT CHR$(&H98);STRING$(HH,CHR$(&H95));CHR$(
&H99)
1510 HL=H+HH+1
1520 FOR VV=V+1 TO 21
1530 LOCATE H,VV:PRINT CHR$(&H96):LOCATE HL,VV:PRINT CHR$(&H96)
1540 NEXT
1550 LOCATE H,VV:PRINT CHR$(&H9A);STRING$(HH,CHR$(&H95));CHR$(&H9B)
1560 LOCATE H+10,V+2:PRINT"<<< Menu >>>":H=H+6:C=1
1570 FOR VV=V TO V+2*5 STEP 2
1580 LOCATE H,VV+5:PRINT C;"... ";MS$(C):C=C+1
1590 NEXT
1600 LOCATE H-1,VV+6:PRINT"Select(1-6)? ";
1610 KY$=INPUT$(1):KY=VAL(KY$):IF KY<1 OR KY>6 THEN BEEP:GOTO 1610
1620 PRINT MS$(KY):GOSUB *BUZZ
1630 RETURN
1640 '
1650 CLS:LOCATE 33,15:PRINT"Thank you!!":CONSOLE 0,25:END
1660 '---------------------
1670 '---- Error ｼｮﾘ ------
1680 '---------------------
1690 ER=1:PRINT:PRINT:BEEP:IF ERR<>53 THEN 1710
1700 PRINT"<<<< ";:COLOR 2:PRINT FI$;" ﾌｧｲﾙｶﾞ ｱﾘﾏｾﾝ!!";:COLOR 0:PRINT" >>>>":GOT
O 1730
1710 IF ERR<>68 THEN 1740
1720 PRINT"<<<< ":COLOR 2:PRINT"Disk ｶﾞ ｲｯﾊﾟｲﾃﾞｽ !!":COLOR 0:PRINT" >>>>"
1730 FOR T=1 TO 3000:NEXT:RESUME 1630
1740 ON ERROR GOTO 0
1750 '---------------------
1760 '---- ﾏｼﾝｺﾞ ｶｷｺﾐ -----
1770 '---------------------
1780 RESTORE:AD=&HE500:CONSOLE 0,25,1,0
1790 LOCATE 30,10:COLOR 2:PRINT"ﾏｼﾝｺﾞ ｶｷｺﾐﾁｭｳ !!":COLOR 0
1800 READ D$:IF D$<>"end" THEN POKE AD,VAL("&H"+D$):AD=AD+1:GOTO 1800
1810 CLS:RETURN
1820 DATA 3E,89,D3,A3,D3,B3,3E,FF,D3,B0,C9,F3,32,AF,E5,D3
1830 DATA B0,3E,89,D3,A3,06,00,3A,AF,E5,CB,FF,D3,B0,7B,D3
1840 DATA A0,7A,D3,A1,3A,AF,E5,CB,BF,D3,B0,DB,A2,DB,A2,DB
1850 DATA A2,77,13,23,10,E1,0D,20,DE,CD,06,E5,FB,C9,F3,32
1860 DATA AF,E5,47,3E,80,D3,A3,78,CB,FF,D3,B0,CD,77,E5,06
1870 DATA 00,7B,D3,A0,7A,D3,A1,7E,D3,A2,3A,AF,E5,CB,87,D3
1880 DATA B0,00,00,00,CB,C7,D3,B0,23,13,10,E5,0D,20,E2,CD
1890 DATA 06,E5,CD,7E,E5,FB,C9,3A,C2,E6,F6,06,18,05,3A,C2
1900 DATA E6,E6,F9,D3,31,32,C2,E6,C9,06,01,18,02,06,FF,0E
1910 DATA 40,3E,9F,18,0A,06,01,18,02,06,FF,0E,20,3E,EF,11
1920 DATA 00,00,21,00,40,10,04,CD,0B,E5,C9,CD,3E,E5,C9,00
1930 DATA F1,21,D3,E5,11,A0,F1,01,0C,00,ED,B0,AF,32,CF,F1
1940 DATA 32,DE,F1,32,F4,F1,3E,48,32,CE,F1,3E,03,32,E1,F1
1950 DATA C3,43,60,F3,3A,C2,E6,F6,02,D3,31,7E,C3,6C,F1,00
1960 DATA "end"
1970 '=======================
1980 '         TAKO9.7
1990 ' Famicom Back up Program
2000 '  Copyright 1985.8.29
2010 '     by Saichi-Tanji
2020 '=======================
```

をコネクタに差し、5文字以内のファイル・ネームを入力すれば自動的に入れてくれます。SAVE時に限らずLOAD時などのファイル・ネームの入力時には、9表に示すように入出力先指定とファイルスの表示が可能です。(カセットのセーブは約7分かかります。またディスク1枚に6本のゲームが入ります)

①Auto Load(オートロード)モード

ディスクやカセットに入っているバックアップソフトをRAMカードに書き込むモードです。5文字以内のファイル・ネーム入力後、オートマチックで書き込んでしまいます。(ディスクからのロード時間は約40秒です)

RAMに書き込んでいる時は、N88BASICが切り離されますから、N88BASICを選択しているときに光っているLEDが消えるので、その確認ができます。

ロードが終わったら、RAMカードをぬき、(Write SwをOFF)にして、ファミコンに差せば一般のROMカセット同様に楽しめます。

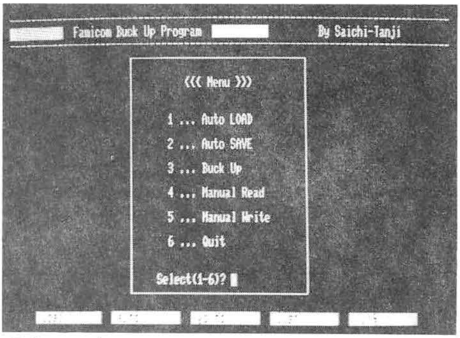
③Back up(バックアップ)モード

このモードは、ROMカセットの内容をソクRAMカードに書き込んでしまうモードです。これを使えばちゃんと動作できるかがすぐ分かります。

操作方法は、ROMカセットを入れ何かのキーを押す。次にRAMカードを入れ何かのキーを押すというように交互にぬき差ししてRAMカードに書き込むものです。

④Manual Read(マニュアル・リード)モード

このモードは、いろいろなメディア(ROM、RAMカード、ディスク、カセット)の中から、プログラムやキャラクタのみをロードして、88のメモリに書き込んでしまうものです。書き込まれるメモリは4000H〜7FFFH(キャラクタは5FFFH)番地ですから、リード後、モニタに落して修整したり、



〈写真43〉tako 88を実行させたところ、メニュー画面

内容を見ることができます。

リードするメディアと、読むファイル名の入力をします。次にリードする内容を指定すれば88のメモリ上に読み込んでくれます。

⑤Manual write(マニュアル・ライト)モード

このモードは、④の逆動作を行なうもので88のメモリ上(4000〜7FFFH)のものをRAMカードなどのメディアに書き込むものです。(操作は同じ)

④と⑤を利用すれば、ゲームのキャラクタを④で読み、別のゲームが入っているRAMカードのキャラクタ部に⑤を使って書き込んでしまうとゲームは同じでも別のキャラクタで遊べてしまうのです。(ちゃんとした形になるかは保障しません!)また、④で読んだ内容を、モニタに落してメモリ内容を修整し、それを⑤を使って書き込んでしまう、なんていうことも可能です。

⑥Quit

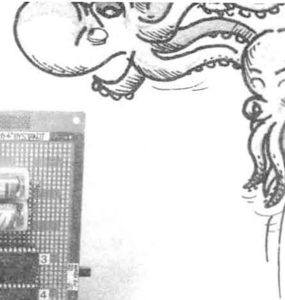
このソフトを終了します。STOPキーやこのモードで終了しても4000H〜に入っている内容はこわされません。何度でもRUN可能です。

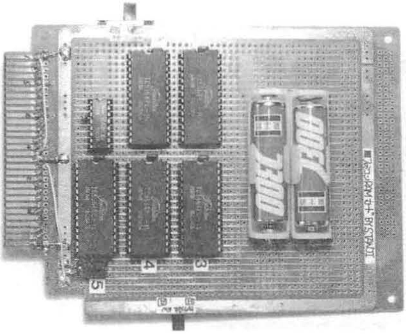
解析できないものもあった

この"タコ吸い"を使っても100%ROMカセットの内容をバックアップすることはできませんでした。いろいろとタイミングなどを

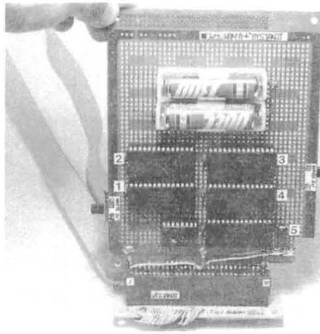








〈写真44－A〉〝RAMカード〟完成写真、おもてより



〈写真44－B〉コネクタにRAMカードを入れる

〈9表〉**ファイル・ネームの入力時には**

●ファイルネームとしてＸＥＶを入力すると
　ＸＥＶ1.Pro ←―プログラムの前半（メモリアドレスで0000～3FFF）⇨88内の4000～7FFFに入る
　ＸＥＶ2.Pro ←―プログラムの後半（メモリアドレスで4000～７ＦＦＦ）⇨　〃　　〃
　ＸＥＶ.chr ←―キャラクタデータ（メモリアドレスで0000～1FFF）⇨4000～5FFFに入る
　の３つのファイルが作られるが入力はすべてＸＥＶでＯＫ！
●このモードをぬけたい！
　⏎のみを押す（ファイルネームなどを押さず⏎キーのみ押す）
●カセットに対する入出力は？
　CAS:XEVでカセット動作となる。
●ディスクのドライブ２を使用したいとき
　２：XEV（２：を頭に付ける）
●ファイルスを見たい！
　FILES（ドライブ１）、FILES２（ドライブ２）を実行

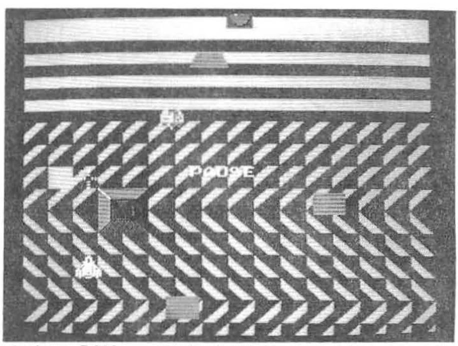
変えて実験したのですが､うまく取れないものは何度やってもだめでした。

まあっちょっとしたミソがあるのではないかと思いますが､もうちょっと研究して100%にしてみたいと思ってます。10表にバックアップしてみた結果をまとめてみました。もしこれ以外の部分で結果がわかりましたら、ぜひ編集部までお知らせください。

RAMカードで遊ぶ

ただ単にゲームをバックアップしていたのではちっとも面白くありません。そこでＲＡＭカードを使ったおもしろい遊び方を紹介しましょう。

まず、ＲＡＭカード①のＲＡＭを抜き、ゼビウスをＲＡＭカードに入れて実行してみます。すると、バックの画面は、きれいな模様になってしまいます。でも飛来するものは、まったく変わっておらず同様にゲームが楽し



〈写真45〉RAMカードの①のRAMをぬき、ゼビウスをロードすると、ゲーム・キャラクターは同じだが、バックはきれいな模様になっている。地下要塞も四角く抜けているのですぐわかる。この例ではピラミッドの左上。

〈10表〉吸い出しバックアップ・リスト

動作状態は
- ◎：完動（かくしコマンド、デモなどすべてＯＫ）
- ○：完動かチェック未完（見た目は完動）
- △：場合によっては、完動（読むタイミングが悪いとダメ）
- ×：何回やっても動作しない

ＲＯＭカセット名	販売会社	動作	キャラジェネスイッチ	備　考
エキサイトバイク	任天堂	◎	H	デザインもＯＫ
ゼビウス	ナムコ	◎	A	死なないソルバルウ　モードもＯＫ！
スペースインベーダパート２	タイトー	◎	H	
ドアドア	エニックス	×	—	表示のみで動作せず
ドルアーガの塔	ナムコ	◎	H	裏ドルアーガもＯＫ
テニス	任天堂	◎	H	
チャンピオンシップロードランナー	ハドソン	◎	H	
エレベータアクション	タイトー	◎	A	
スターフォース	ハドソン	×	—	表示もされず
ギャラクシアン	ナムコ	◎	A	
ベースボール	任天堂	◎	A	
ロードファイター	コナミ	◎	A	
Ｆ１レース	任天堂	◎	H	
マッピー	ナムコ	◎	H	
ジッピーレース	アイレム販売	◎	A	
ハイパーオリンピック	コナミ	○	A	専用のジョイパッドがなかったのでゲームは試していないが、デモなどを見る限りＯＫのようだ。
ドンキーコング	任天堂	△	H	何回かに１回はちゃんと動作する。動作したときのデータを保存しよう
ピンボール	任天堂	◎	H	
エクセリオン	ジャレコ	◎	A	
ホーガンズアレイ	任天堂	○	H	ピストルがなかったので、ゲームを試すことができず完動かは？　見た目はＯＫ！
スパルタンＸ	任天堂	×	—	表示もされずダメ（キャラクタはＯＫ）

めます。その上、今まではよく分からなかった地下要塞の位置が四角くきれいに抜けているので、そこを攻撃すれば必ず地下要塞が出て来ます。（写真45）

ＲＡＭを元に戻して、ゼビウスのプログラムに、スパルタンＸのキャラクタを入れると、バックの地形にラーメンカーブが表示され、赤い回転灯をチカチカさせて飛んでくるコーヒーカップがお目にかかれます。

などなど、いろいろ組み合わせて遊ぶととても、おもしろいですよ。

〝ファミコンのゲーム・カセット〟をいっぱい持っている人と友達になったほうが良さそうですね。



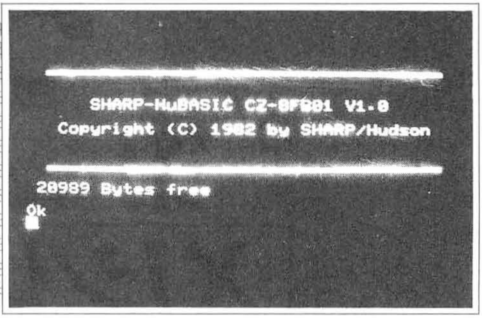

第3章

X1シリーズ 徹底内部解析&バックUPテクニック

多摩 三郎



Z－80の知られざる世界



●X1流Z－80の究極の使い方

〝究極の8ビット（PC-8801)〞とか〝究極の8ビットCPU（6809)〞とか究極と言う言葉もずいぶん使い古されましたが実は、Z-80CPUにも究極の使い方があったのです。

ここで読者の皆さんの常識をくつがえす重大な事実を申し上げます。

> Z-80が一度にサポートできるメモリ空間は128Kバイトなのだ!!

ということです。今まではバンク切り替えも何もしなければ、Z-80は64Kバイトのメモリしかサポートできないというのが常識でした。では残りの64Kバイトは、一体どこにあるのでしょうか？？

それは、I/Oポートにあるのです。Z-80は8ビットCPUですからアドレス・バスは16本ありますので、全部で64Kバイトのアドレスを指定できます。(図1)

一方、I/Oポートは同じアドレス・バスを使いながら、普通はもったいないことに8本しか利用していませんので、最大でも256個のI/Oポートしか指定できないことになっています。つまり、残りの8本は利用されずにいたわけです。

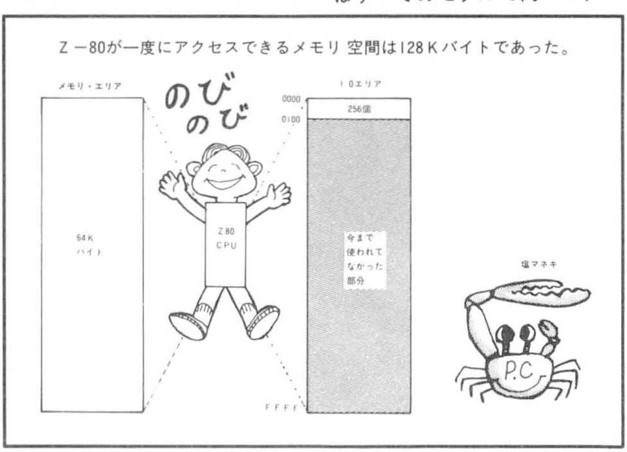
この未使用のアドレス・バスをすべて使えば全部で65536個のI/Oが使えるようになり、ここにメモリを割り付ければZ-80は新たに64Kバイトのメモリ空間をサポートできるようになる、という訳です。

どうですか、これがX1流Z-80の究極の使い方なのです。(究極の8ビット機は、X1ターボでしたが最近MZ-2500にのっとられたようだ。)

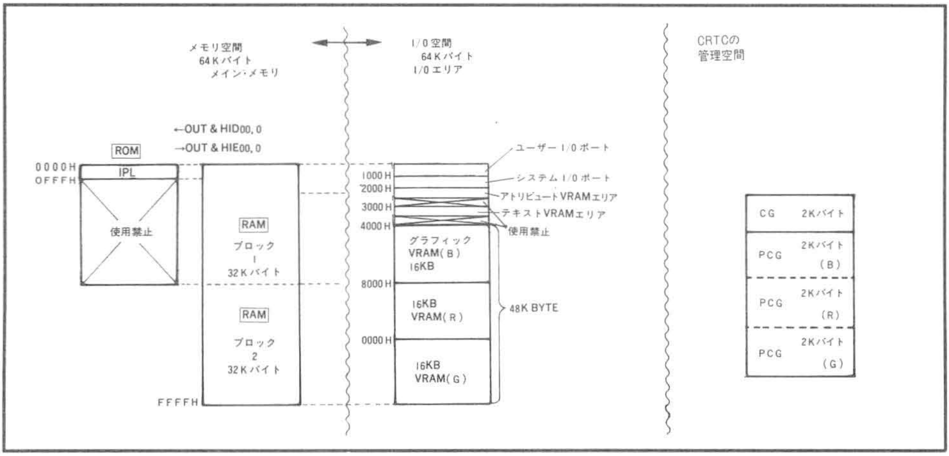
●X1のメモリ・マップ

X1では、このように究極的なZ-80の使い方のおかげで、大容量のメモリも割合にゆったりと使うことができます。機能的には640×400ドットのモノクロができない他はPC-8801mkIIとほぼ同等でありスピードはFM-7といい勝負なのですから、これはエライことです。

X1には多くのモデルがありますが、元祖X1（CZ-800C）でオプションだったものが、後のモデルで標準装備されて安くなった他は大きな変化はありません。従って、メモリ・マップはすべてのモデルで同一で、ソ



〈図1〉 Z－80の知られざる世界



〈図2〉 X1のメモリ・マップ

フトもフルコンパチです。

あれだけデザインが変化したのに中身はまったく同じというのは、他に例がありませんし、少々信じがたいことでしょうが、事実は事実です。最近発売されたX1FにはNEW-BASICというソフトが付いてきますが、元祖X1のユーザーもこのBASICを買って走らせれば、X1Fに変身してしまうのです。後で紹介するX1ターボのコンパチビリティも半ぱではありませんよ。

それでは実際にX1のメモリ・マップを見ていくことにします。

X1は64Kバイトのメイン・メモリ（すべてRAM）と48KのグラフィックV-RAM（I/Oアドレスの4000H～FFFFH）、4KバイトのIPL ROM、CRTCの管理下にあるPCG（プログラマブル・キャラクタ・ジェネレータ。MSXのスプライトのようなもの）用として6KバイトのRAMがあります。（図2）

割合すっきりしているのは、Z-80の究極的使用法のおかげです。48Kバイト3階建てのグラフィックRAMというのは、一見FM-7にも似ています。また、64Kバイト内のどのアドレスからでもバンクの切り替えなしで、グラフィックV-RAMをアクセスできるところは、ゲームを作る側にとって有難いことです。

今年の8月に発売されたNEW-BASICの超々高速ペイントを見ていると、さすがにバンク切り替え不要の効果は大きいようです。（旧BASICのペイント文のトロさは、一体何だったのでしょう）

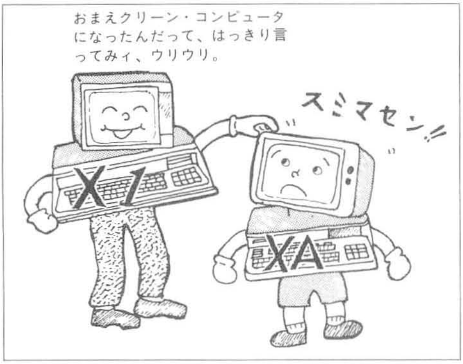
X1はオールRAMですからBASIC ROMというものは存在しません。電源を入れるとIPLの働きでBASICやゲームがメイン・メモリに読み込まれます。ディスクが普及してきたので"クリーン・コンピュータ設計"というのも、あながち不利とも言い切れませんね。（PC-98XAだってクリーン・コンピュータなのですよ！）

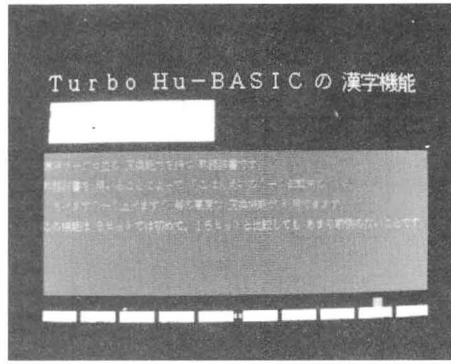
IPLはInitial Program Loaderと呼ばれ、プログラムを最初に読み込むためにあるもので、必要最少限の機能しか備わっていません。もちろんROMでなければ、お話しになりませんが。

X1では、このIPLが、なかなかのスグレモノなのです。OUT & H1D000、A（Aの内容は何でも良い）とすると、IPL ACTIVEの状態になります。この状態はバンク切り替えではなく、ちょうどPC-8801のBASIC ROMのように0000H～0FFFHを読むとROMの内容が読み出され、このアドレスに書き込むとRAMに書かれるようになっています。これを一般にシャドウRAMと読んでいます。

また、PC-8801などでは長いマシン語プログラムをディスクに転送して起動しようとすると、RAMの真ん中あたりに転送されているワーク・エリア（ディスク・コードと言う）を壊して立ち上げられないことがありますが、X1のIPLではすべての処理をROM内で行なうため、サンダーフォースのような長いプログラム（0000H～FEFFHまでロードする）でも一発で立ち上げることができるのです。

＊CRTC／カラー信号の発生、同期信号の発生、その他画面表示に必要な処理一切を受け持つIC。





Ｘ１ターボは漢字処理が強力だ!!

●Ｘ１ターボのメモリ・マップ

Ｘ１ターボは、元祖Ｘ１との完全な互換性をたもちながら大幅な拡張をして16ビット・マシン並の性能を持たせてしまうという過激な目標の上に作られたパソコンです。

PC-8801mk II SR でも果せなかった640×400ドットのフルカラーを実現するために、グラフィックＶ-ＲＡＭの容量が倍の96Ｋバイトに拡張されており、さらに16ビット機の PC-9801 で初めて採用され好評だった漢字テキスト-ＲＡＭ(漢字をグラフィック画面に表示するのではなく、ＡＢＣ等の文字を表示する画面に表示する方法)を８ビット機では初めて実現しています。

また、ＩＰＬ　ＲＯＭを大幅に拡張しBIOS ROM（32Ｋバイト)としています。BIOS ROM は BASIC をはじめとし、あらゆるプログラムで必ず利用されると思われる基本的な部分、例えばキー入力や画面表示、ディスク入出力、演算ルーチンや LINE、ペイントなどで必要とされる部分だけを集めてＲＯＭ化したもので、いわゆる BASIC ROM とは違うものです。（図３）

あえて表現するならば、CP/M やＭＳ-ＤＯＳなどのＯＳ(オペレーション・システム）をＲＯＭ化したものと考えれば良いでしょう。なんとキー入力ルーチンをコールするだけで漢字辞書までサポートしてくれるのです。また、メモリの前半の部分００００Ｈ～７ＦＦＦＨまでを大胆にも32Ｋバイト単位でバンク切り替えすることで、512ＫバイトものＲ A Ｍ増設を可能にしています。

BASIC はＲＡＭブロック１（００００Ｈ～７ＦＦＦＨ）の部分に読み込まれ、BIOS ＲＯＭと重なりますが、シャドウＲＡＭとＲＳＴ 18Ｈをうまく使うことで64Ｋバイト内のどのアドレスからでも BIOS ROM を利用できるようになっています。

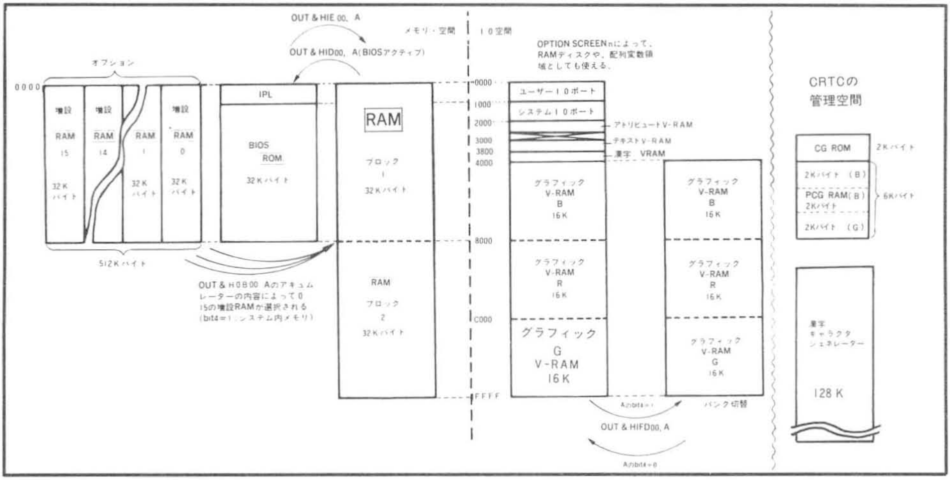
Ｘ１のアドベンチャー・ゲームが少なかったことの原因の１つには、旧 BASIC のペイント文のトロさもあったのでしょうが、それより大きな原因はクリーン・コンピュータだったということでしょう。

アドベンチャー・ゲームを制作する側にとって、比較的力のないソフト・ハウスにおいてはオリジナルの高速ペイント・ルーチンを開発することは、容易ではありません。ROM BASIC のＰＣやＦＭでは、BASIC 内のペイント・ルーチンやディスクＩ/Ｏを借りてこれるので良いのですが、クリーン・コンピュータではすべて自作しなければならないからです。

実はシャープと契約を取り交わせば、BASIC を無償で商品に組み込んで販売することも可能ですが、旧 BASIC のペイントではとても使いものになりません。従ってＸ１のアドベンチャーは後回しということになるのです。

この点ターボでは、必要なルーチンがすべて BIOS ROM 内にあるため簡単に利用できます。〝アステカ〟や〝王家の谷〟など640×400ドットのグラフィックでもないのにターボ専用のソフトが出ていることを見ると、あながち外れているとも思えません。ターボの BIOS*[1] ROM は BASIC でもない、クリーンでもない、一番おいしいところだけを採用しようとした末の結論なのでしょうか。

ところで、ターボはこれだけの拡張をしておきながら、元祖Ｘ１のメモリ・マップをソック



〈図3〉 X1ターボのメモリ・マップ

り含んでいるために、ほぼ完全な上位コンパチを保っています。X1ターボには、いわゆるモード切り替えというものがありません。前面パネルのモードを200ラインとしておけば、ほとんどすべての（X1用の）ソフトが走りますが、決して元祖X1になったのではなく、あくまでターボはターボのモードのままで元祖X1のソフトが走っているわけです。

こういったことは、コンピュータ史上初めてのできごとです。巷ではモード切り替えという名のもとに、複数のコンピュータを1台の中に抱き合わせ、スイッチで切り替えて使っているのが現状ですが、長期的視野に立って慎重に設計を進めないとなかなかこうは行かないものです。

《64バイトのI/Oポートのアクセス》

64KバイトものI/Oポートをアクセスするには、BCレジスタを用いた間接アドレッシングというテクニックを駆使します。

いきなりマシン語の話になってしまいましたが、X1のプロテクトの解析を進めていくと必ずといって良いほどぶつかる問題ですし、I/Oポートの16ビット・アドレスの制御の仕方というのは、どんなマシン語の解説書にもほとんど載っていないことですので、マシン語のわかる方はゼヒ頭の中に入れておいてください。

例えばI/Oポート4000Hをアクセスするには次のようにします。

```
LD   BC、4000H
OUT  (C)、A
```

また、同じようにして

```
INA (C)
```

も使うことができます。これは16ビット・アドレスの上位8ビットにBレジスタの内容が出ているという性質を利用したものです。また、アキュムレータの内容はどうでもよく単にアクセスするだけで良いという場合は、

```
LD A、1DH
OUT (00H)、A
```

というようにすれば、アドレスの上位8ビットにアキュムレータの内容が出ますので、1D00Hがアクセスされます。

また、

```
LD  A、1AH
IN  A、(01H)
```

とすると、アキュムレータにはI/Oポート1A01Hの内容が入ります。

これらはカセット入出力ルーチンなどで使われることが非常に多いので、注意していてください。

※1 なお、BIOSをROMで持つという方法はMZ-2500でも採用されています。OSのROM化という考え方はこれからはやるのではないでしょうか。

X1全機種の性能・機能比較



一口にX1といっても、たくさんのモデルがありますが、大きく分ければX1シリーズとターボに分かれます。

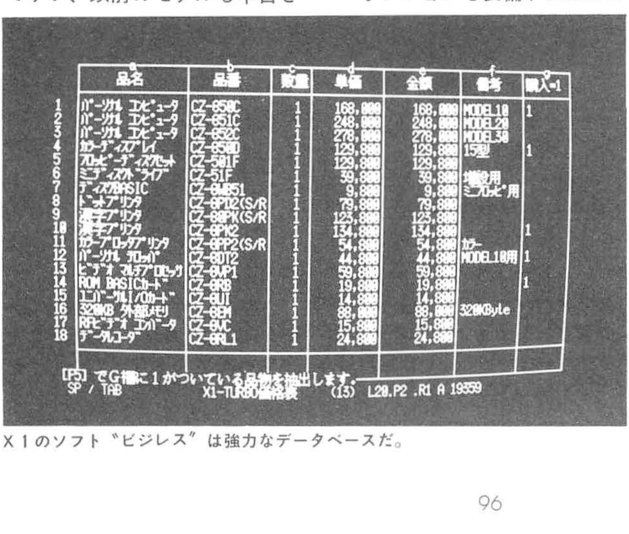
X1シリーズには、元祖X1(マニタ・タイプ)を始めとしてC/D/Cs/Ckと今年7月に発売のFがあります。基本的なハード・ウエアは全く同じ(X1Dでテープを使う場合を除く)ですから、あるソフトがどの機種で動いていて、どの機種で動かないとかいったことはありません。

X1Fに標準装備のNEW BASIC(ペイントの速さがFM-77並になった)も全機種で走ります。また、ソフトも増えていますし、〝サンダーフォース〞や〝プラズマライン〞など、X1でなければお話にならないようなソフトも出てくるようになったところですから、そろそろここいらでセカンド・マシンとして購入してみようかと思っていらっしゃる方も多いのではないでしょうか。

●お買い得なX1の機種選び

新品を購入するのなら、新発売のX1Fモデル20(定価139,800円)が何よりもお買い得ですが、以前のモデルも中古として随分安く出回るようになりましたから、賢い中古作戦を立てるのも良いかもしれません。

オプションを装備すればどのモデルもX1Fと同じ仕様になるのですが、組み合わせにより有利不利がありますので、オプションの価格とモデルの関係を表1にまとめてみました。



	品名	品番	数量	単価	金額	備考	購入=1
1	パーソナル コンピュータ	CZ-850C	1	168,000	168,000	MODEL10	1
2	パーソナル コンピュータ	CZ-851C	1	248,000	248,000	MODEL20	
3	パーソナル コンピュータ	CZ-852C	1	278,000	278,000	MODEL30	
4	カラーディスプレイ	CZ-850D	1	129,800	129,800	15型	1
5	フロッピーディスクセット	CZ-501F	1	129,800	129,800		
6	ミニディスクドライブ	CZ-51F	1	39,800	39,800	増設用	
7	ディスクBASIC	CZ-8WB51	1	9,800	9,800	ミニフロッピー用	
8	ドットプリンタ	CZ-8PD2(S/R	1	79,800	79,800		
9	漢字プリンタ	CZ-80PK(S/R	1	123,800	123,800		
10	漢字プリンタ	CZ-8PK2	1	134,800	134,800		1
11	カラープロッタプリンタ	CZ-8PP2(S/R	1	54,800	54,800	カラー	
12	パーソナル テロッパ	CZ-8DT2	1	44,800	44,800	MODEL10用	1
13	ビデオ マルチプロセッサ	CZ-8VP1	1	59,800	59,800		
14	ROM BASICカード	CZ-8RB	1	19,800	19,800		1
15	ユニバーサルI/Oカード	CZ-8UI	1	14,800	14,800		
16	320KB 外部メモリ	CZ-8EM	1	88,000	88,000	320KByte	
17	RFビデオ コンバータ	CZ-8VC	1	15,800	15,800		
18	データレコーダ	CZ-8RL1	1	24,800	24,800		

[F5] でG欄に1がついている品物を抽出します。
SP / TAB　X1-TURBO価格表　(13)　L20.P2 .R1 A 19359

X1のソフト〝ビジレス〞は強力なデータベースだ。

オプションは、この他にもいろいろありますが、もしかすると次のモデルでは320K外部RAM(CZ-8EM 定価88,000円)やパーソナル・テロッパー(CZ-8DT2 定価44,800円)などが標準装備になっているかも知れません。フル装備の合計価格を見ていると、この3年間にいかにハード・ウエアが安くなってきたかが良く分かると思います。

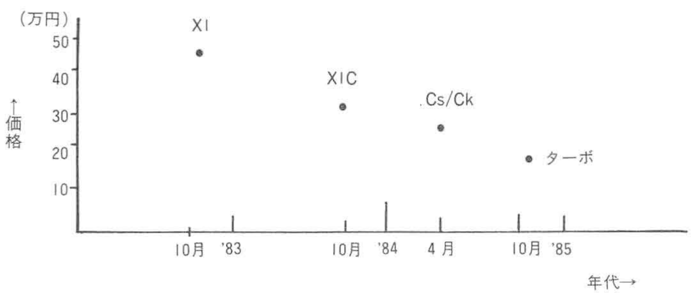
また、この中古価格がどのくらいであれば、お買い得かということが計算できます。X1Fモデル20の199,400円を基準にすれば、例えば元祖マニア・タイプが199,400円—8,800円(NEW BASIC)—129,800円(ディスク・ユニット)—19,800円(漢字ROM)—11,800円(拡張I/Oポート)=29,200円

つまり、G-RAM付きで29,200円以下なら、これはお買い得ということになるわけです。(いくら中古でも29,200円でX1が買えるわけがない!)

それにしてもX1Fモデル20は安い!! こうなるとCタイプのユーザーなんか、199,400円—

〈表１〉フル装備の合計価格とX1のオプション価格表

オプション（標準装備は○）	G-RAM	電磁メカカセットレコーダー	拡張I/Oポートまたはボックス	漢字ROM	フロッピーディスク				フル装備合計価格
					インターフェース	ディスクユニット(台)	増設ユニット	NEW BASIC	
X1マニアタイプ CZ-800C ¥155,000	¥32,000 CZ-8GR	○	¥11,800 CZ-8EP	¥19,800 CZ-8BK2	FDDに同梱	¥129,800 (2) CZ-501F	――	¥8,800 CZ-124SF	¥470,200
C CZ-801C ¥119,800	○	○	¥29,800 CZ-81EB	¥19,800 CZ-8BK2	〃	¥129,800 (2) CZ-501F	――	¥8,800 CZ-124SF	¥308,000
Cs CZ-803C ¥119,800	○	○	○	¥19,800 CZ-8BK2	〃	¥129,800 (2) CZ-501F	――	¥8,800 CZ-124SF	¥278,200
Ck CZ-804C ¥139,800	○	○	○	○	〃	¥129,800 (2) CZ-501F	――	¥8,800 CZ-124SF	¥278,400
D CZ-802C ¥198,000	○	¥24,800 CZ-8RL1 (マニアル操作のみ)	¥11,800 CZ-8EP	¥19,800 CZ-8BK2	○	○ 3インチ(1)	¥59,800 CZ-31F	¥8,800 CZ-113SF	¥323,000
F model 10 CZ-811C ¥89,800	○	○	○	¥19,800 CZ-8BK2	FDDに同梱	¥129,800 ○(2) CZ-501F	――	¥8,800 CZ-124SF	¥248,200
F model 20 CZ-812C ¥139,800	○	¥24,800 CZ-8RL1 (フルロジック)	○	○	○	○ 5インチ(1)	¥34,800 CZ-52F	○	¥199,400



8,800円（NEW BASIC）－129,800円(ディスク・ユニット)-19,800円(漢字ROM)-29,800円（拡張I/Oボックス)＝11,200円

何と11,200円以上で中古に出せるのなら、オプションを揃えるより買い替えた方がお得ということになってしまいます。その上、Cタイプのキーボードの質は最悪でカセット・メカもすぐ壊れるし……。

Dタイプは、ソフトによるAPSSができないので、テープ版のソフトの一部（〝ゼビウス〟等）に問題がでることがあります。X1DでもAPSSが使えるように改造するという記事が、マイコン（1985年9月号）に載っていました。腕に覚えがある人は挑戦してみると良いでしょう。まあディスクだけを使っている分には全く問題がありません。3インチのディスクがマイナーになってしまったので、5インチのディスクを増設した方が良いかもしれません。

X1ターボにはモデル10からモデル40までの4モデルありますが、このうちモデル10から30までは、オプションをフル装備すれば全く同じになります。

モデル10は一見して安そうですが、G-RAMの1ページとRS-232C/マウス・インターフェイスはオプションですし、まさかターボ専用のテープ版ソフトというのはまず発売されないでしょうから、結局ディスクを購

〈表2〉 X1ターボのオプション価格表

¥258,000 X1ターボ Model 40	¥278,000 X1ターボ Model 30	¥248,000 X1ターボ Model 20	¥168,000 X1ターボ Model 10			
○	○	○	¥14,800 CZ-8GR2	増設G-RAM ページ2		
○(2)	○(2)	○(2)	○(3)	I/Oポート(数)		
○	○	○	○	高速漢ROM (漢字キャラクタジェネレーター)		
○	○	○	FDDに同梱	2D	インターフェース	フロッピー・ハードディスク
			(2D～2HD自動切替用インターフェース)予想	2DD 2HD	インターフェース	フロッピー・ハードディスク
¥30,000 PIO-4100 (I・0データ機器)	¥30,000 PIO-4100 (I・0データ機器)	¥30,000 PIO-4100 (I・0データ機器)	¥30,000 PIO-4100 (I・0データ機器)	8インチ	インターフェース	フロッピー・ハードディスク
¥18,000 KGB-HDIF (計測技研)	¥18,000 KGB-HDIF (計測技研)	¥18,000 KGB-HDIF (計測技研)	¥18,000 KGB-HDIF (計測技研)	ハードディスク	インターフェース	フロッピー・ハードディスク
○(2)	○(2)	¥39,800 (1) CZ-51F	¥129,800 (2) CZ-501F	2D	ディスクユニット	フロッピー・ハードディスク
未発売	未発売	未発売	未発売	2DD 2HD 切替	ディスクユニット	フロッピー・ハードディスク
¥208,000 PFD-8E (I・0データ機器)	¥208,000 PFD-8E (I・0データ機器)	¥208,000 PFD-8E (I・0データ機器)	¥208,000 PFD-8E (I・0データ機器)	8インチ	ディスクユニット	フロッピー・ハードディスク
¥268,000 M-510 (アイテム)		¥288,000 PC-98H33K (NEC)		ハードディスク	ディスクユニット	フロッピー・ハードディスク
¥24,800 CZ-8RL1 専用ケーブル ¥1,400	¥24,800 CZ-8RL1	¥24,800 CZ-8RL1	○	電磁メカカセット		
○	¥8,800 CZ-111SF	¥8,800 CZ-111SF	¥8,800 CZ-111SF	システム(熟語)辞書		
増設不能	○	○	¥44,800 CZ-8DT2	SSS(スリーエス)方式テロッパー	AV機能	
	○	○	○	テレビコントロールスーパーインポーズなど	AV機能	
○	○	○	発売予定	RS-232C & マウスインターフェース		

入することになります。

これだけでもモデル30の価格を上回るというのに、2Dディスク・ユニットに同梱のインターフェイスは近日発売の2HDユニットや将来発売される2DD・2HD自動切り替えユニットには対応していませんので、将来の出費はさらに大きくなります。

また、いくら必要を感じないからといっても、放送局並の性能を持つというテロッパー(モデル20、30に標準)を無視するわけにもいかないでしょう。

問題はモデル40ですが、モデル30でオプションだったシステム・ユーザー辞書を同梱しただけのことで、これで日本語処理機能を強化したとはお笑いです。その上、カセット・ケーブルやRGBケーブルは別売(セコイ!)です。

また、何よりも肝心なのはAVコントロール機能がなく増設もできないので、将来マイコンとビデオ・ディスクをつなげて迫力のビデオ・ゲームを楽しむ道も、とざされてしまうことです。これではX1ではない!!仲間など入れてやるものか。

モデル40の噂が流れたとき、性能が良くて安いモデルが出るというので、名古屋の業者が在



庫のターボを叩き売ったことがありましたが、早合点して丸ゾンした業者のおかげで、たくさんのユーザーが得をしました。モデル40でも、たまには役に立つことがあったんですね。

結局モデル30がよろしいようで……。

可能である点など、一般の16ビット機より優れているというのに、8ビットだからビジネス用に使えないとは言わせません。

結局、X1ターボというのは、何に使っても優秀なのです。豪華けんらんで機能は満載、X1ターボは史上初めてのオールマイティ・パソコンです。かつてワンボード・マイコンが登場したとき、誰もが描いた自然な夢をX1ターボは実現しようとしています。

●X1ターボはオールマイティ??

X1ターボには〝ゼビウス〟や〝サンダーフォース〟などのスグレモノ!!ゲームがあり、ジョイスティックが2本も使えます。これだけでゲームに最高といえます。

ビデオ編集にテロップが入れられて、テロップ用の文字としてMSX並のデカ文字(40字×10行)が使えます。AV(オーディオ・ビジュアル)はX1の独断場です。高解像度モードにすればPC-9801並の高解像度グラフィックス(640×400ドット・フルカラー)に高速ペイント、マウスも使えて評判のグラフィック・ソフト〝Ink Pot〟が走ります。

漢字キャラ・ジェネがあるので、16ビットのワープロ〝即戦力〟や〝漢字マルチプラン〟が移植されました。プリンタ config ができるので、プリンタならNECだろうがエプソンだろうが、ビット・イメージさえ使えれば純正と同じように漢字が打ち出せます。(リスト1参照)

メモリ容量が足りないなんてことは、ありません。漢字マルチプランは64Kバイトの増設メモリ付きですし、必要とあらば512Kバイトまで増設が可能です。これとは別に、3.2メガ・バイトまでの外部メモリを BIOS がサポートしています。

5インチ・ディスクなら2HDまでのインターフェイスは内蔵で8インチやハード・ディスクも BIOS でサポートされています。当然、BIOS でサポートされているのなら BASIC でも使えますから安心です。

INPUT 文実行中に熟語辞書が使え、ワープロ並の入力が、

●BASICを解剖してみる

アドベンチャー・ゲームや、シュミレーション・ゲームなど、それほどスピードを要しないゲームの中には BASIC で組まれているものもたくさんあります。それらのほとんどがプロテクトがかけてあり、容易にブレークやリストが取れないことがあります。

X1の BASIC は、RAM BASIC ですから、ディスクやテープからRAM上に読み込んで使いますが、オールRAMですから好きなように書き換えることができます。

例えば、ブレークやLISTを使えなくしたり、テープ・フォーマットを変えたり、テキスト(プログラム)の始まる番地を変えたりなど、UNLIST 化やプロテクトの方法は星の数ほど(メモリの数ほどかな?)あるといっても過言ではありません。

X1の BASIC には、元祖旧-BASIC(CZ-8C(F)B01 V1.0)と NEW BASIC(CZ-8C(F)B01 V2.0)、そしてターボ用の漢字 BASIC(CZ-8F(C)B02)と3系統があり、それぞれにテープ・バージョンとディスク・バージョンがありますので計6種類の BASIC があります。(図4a、b)

基本的な構造は全く同じなのですが、どこをどう書き換えれば UNLIST 化できるかといったことなどは、それぞれの BASIC で皆違います。詳しくは次のところで説明いたします。

NEW BASIC やターボの漢字 BASIC で面白いところは、NEW ON命令によっていらない命令を削り、その分フリー・エリアを増やすことができることです(表3、リスト2、P.119参照)

命令を削る順は、ディスク BASIC とテープ BASIC ではそれぞれ違いますが、例えばテレビ・コントロールやカセット・コントロール(ディスク BASIC の場合)などの命令がまず削られ、最後にはエラー・メッセージや LIST、SAVE まで削除されてしまいます。

しかし、ディスクI/O(ディスク BASIC の場合)またはテープI/O(テープ BASIC の場合)や、超々高速ペイント・ルーチン、漢字入出力ルーチン等は最後の方まで残りますからア

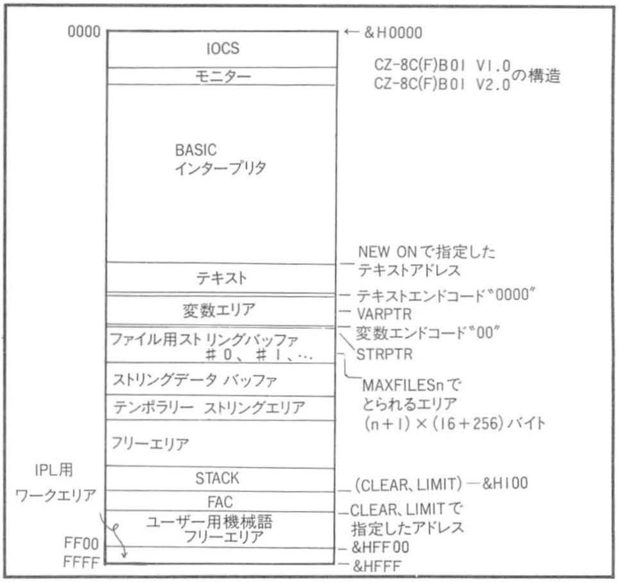
〈リスト 1〉 スターTR-24で打ち出したリスト。"8ビット"ですよ！

```
10 OPTIONSCREEN0:WIDTH 80,25,1,0:INIT:CLS
11 KMODE 1:CONSOLE 0,24,5,72:KBUFON:KEY0,CHR$(&HFF,17,13):A$ INKEY$(1)
12 DIM W(12),X(12),Y(12),Z(12)
15 WINDOW (0,0) (639,399),(0,0) (639,199)
20 COLOR 7:CSIZE 3:LOCATE 22,0:PRINT#0,"2000年の通勤時間は?"
25 KMODE 0:PRINT STRING$(69,"  "):KMODE 1
26 A$ "2000年(昭和75年)の 地域別の 通勤時間距離は どう変化しているであろうか。 まず、三大都市圏では、「長くな
る」とする回答者が 71.4% を占めている(図1)  「長くならない」は28.6%の解答率で、これらの解答者は長くならない
理由として"
27 B$ "「都心部の分離化が進む」などをあげている。                          その他の地域においては、通
勤時間が「長くなる」とした回答者は80.0% と圧到  的な比率である(図2)。「長くならない」とした回答者は20.0% で、こ
れらの回答"
28 C$ "者は長くならない理由として「職住近接が可能」、「交通網の整備」をあげている。"
90 COLOR 5:PRINT A$:PRINT B$:PRINT C$
100 COLOR 7:PRINT:PRINT "(図1)  三大都市圏の通勤時間距離
110 COLOR 4:PRINT "経企庁編" 2000年の日本ｼﾘ ｽﾞ" より
120 LINE (40,100) (600,180),PSET,4,B
130 LINE (40,140) (600,140),PSET,4
140 LINE (170,100) (170,180),PSET,4
150 LOCATE 7,14:COLOR 7:PRINT "長 く な る"
160 LINE (171,104) (480,119),PSET,&H63,BF
170 LINE (171,121) (520,136),PSET,&H75,BF
180 LOCATE 61,13:PRINT "61.5%":LOCATE 66,15:PRINT "71.4%"
190 LOCATE 7,19:COLOR 7:PRINT "長くならない"
200 LINE (171,144) (358,159),PSET,&H63,BF
210 LINE (171,161) (310,176),PSET,&H75,BF
220 LOCATE 46,18:PRINT "36.5%":LOCATE 40,20:PRINT "20.0%"
221 RESTORE 1490
222 GOSUB 1370
223 RESTORE 1500
224 GOSUB 1430
230 FOR I=1TO 20000:NEXT
250
260 WIDTH ,,0:INIT:CLS 4:COLOR 6:CSIZE 3:PRINT#0,"Ｔｕｒｂｏ Ｈｕ  ＢＡＳＩＣ の 漢字機能":LINE (0,65) (63
9,178),PSET,&H10,BF
261 A$="カナ漢字変換 は カナ、ローマ字 のいずれからでも 入力することができます。"
262 B$ "ハードウエアー に ローマ字 変換機能 が 組み込まれておりますので、"
263 C$ "ＢＡＳＩＣからも マシン語からも 簡単に利用することができます。"
264 D$ "辞書 ディスク が 不用、高速です。"
265 LOCATE 34,23:CFLASH 1:PRINT "↓";:CFLASH
270 LINE (8,24) (280,56),PSET,7,BF
280 SYMBOL (8,24),"ローマ字",4,2,2,0,PSET
290 LOCATE 0,9:COLOR 4:PRINT A$
300 FOR I-1TO 10000:NEXT
310 LOCATE 0,9:COLOR 2:PRINT A$
320 LOCATE 0,11:COLOR 4:PRINT B$:PRINT:PRINT C$
330 FOR I 1TO 10000:NEXT
340 LOCATE 0,11:COLOR 2:PRINT B$:PRINT:PRINT C$
350 LOCATE 0,15:COLOR 4:PRINT D$
360 FOR I 1TO 10000:NEXT
370 A$ "Ｘ１ ｔｕｒｂｏ 本体に 組み込まれている、カナ漢字 変換機能です"
371 B$ "この機能も ハードウエアー によって 実行されますので、"
375 LOCATE 0,9:PRINT CHR$(26):LOCATE 52,23:CFLASH 1:PRINT "↓";:CFLASH
380 LINE (8,24) (280,56),PSET,7,BF
385 SYMBOL (8,24),"１字変換",4,2,2,0,PSET
390 LOCATE 0,9:COLOR 4:PRINT A$
400 FOR I-1TO 10000:NEXT
410 LOCATE 0,9:COLOR 2:PRINT A$
420 LOCATE 0,11:COLOR 4:PRINT B$:PRINT:PRINT C$
430 FOR I 1TO 10000:NEXT
440 LOCATE 0,11:COLOR 2:PRINT B$:PRINT:PRINT C$
450 LOCATE 0,15:COLOR 4:PRINT D$
```

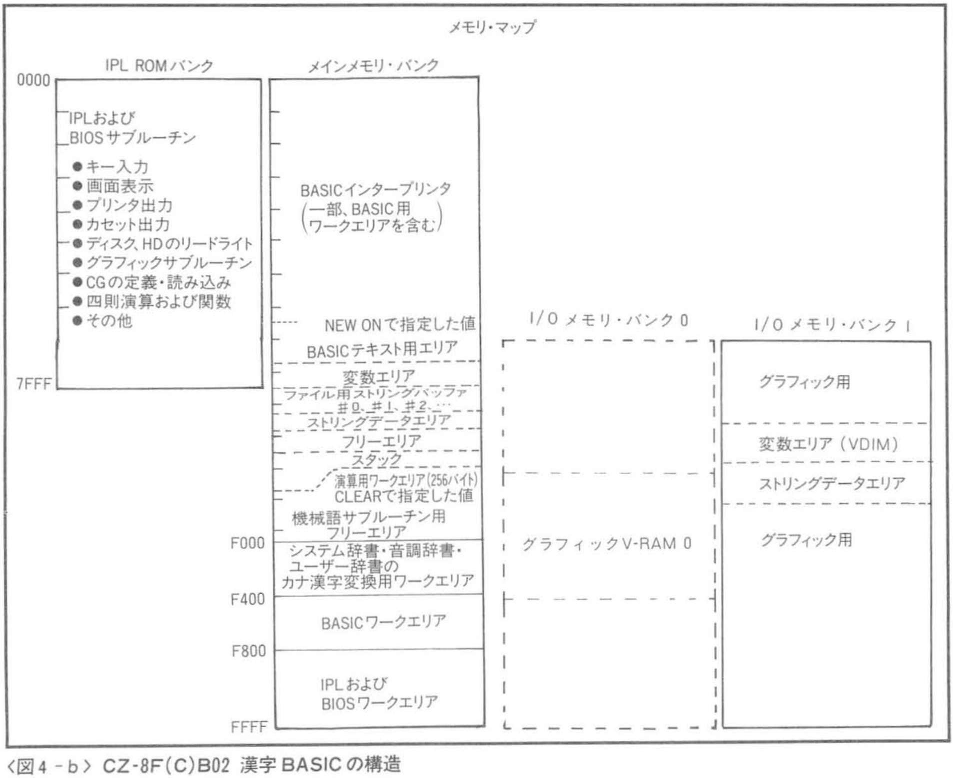
ドベンチャー・ゲームや実務ソフトなどでは困らないようになっています。

また、巷ではシャープはプリンタがガンだ（セントロニクス仕様になっていないから）といわれてきました。しかし、元祖X１以来（もう３年も前のことですよ！）180°方向転換（セントロニクス準拠になった）をしていることをご存じない方も多いようです。

NEW BASICやターボの漢字BASICでは、この考えをさらに進めプリンタConfigというユーティリティを使ってBASIC自身を書き換えて（インストールという）純正以外のプリンタでもハード・コピーや漢字の出力ができるようにしてあります。



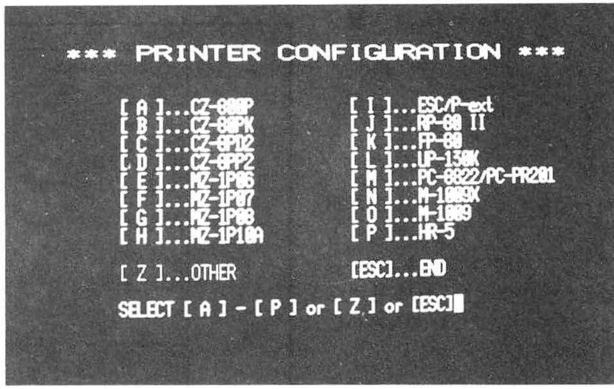
〈図４－a〉CZ-8C(F)B01の構造



〈図４－b〉CZ-8F(C)B02 漢字BASICの構造

プリンタConfigのメニューの中にはＭＺ系のプリンタだけでなく、何と!!ＮＥＣのPC-PR201(24ドット漢字プリンタ)やエプソンのＥＳＣ/Ｐ extまであるのですから、少しやりすぎの感があるくらいすばらしい機能を持たせてあります。

筆者はPC-PR406を使っていますが、ＮＥＣのＣ＆Ｃプラザでこの話をすると、すごくイヤーな顔をされました。でも考えてみてください。ＮＥＣはパソコン・メーカーであると同時に、有数のプリンタ・メーカーでもあるのですよ。どこのマシンにつながっていようと、ユーザーはユーザーなのですから、もっと親切にして欲しいものです!!



プリンタなら何でもＯＫ!!〝プリンタConfigユーティリティ〟

●テープソフトのLoad SaveとIocs

《IOCSとは》

Ｘ１のBASIC(ターボは除く)は、旧BASIC、NEW BASICとも0000Ｈ～149FHまでが共通になっています。このうちの0000Ｈ～0FE1HまでをIOCS(INPUT OUTPUT CONTROL SYSTEMといい、キー入力ルーチンや表示ルーチンを始めとして、カセット入出力ルーチンなど、Ｘ１の制御にかんする大切なサブルーチンがまとめ作り込められている）が占めています。

別名BIOSとかOSとも呼ばれています。IOCSを一ヵ所にまとめて作るのはＭＺ以来の伝統ですが、一ヵ所にまとめておくのはあらゆる点で有利なのです。ハード・ウエアに変更があった際でもIOCSを少し作り替えるだけで対応できるからです。

実際、ＭＺ-2200のBASICの

〈表３〉NEW ONとテキスト・スタート・アドレス

	BASICのプログラム(テキスト)スタートのアドレス(HEX)				
	V1.0		V2.0		
	CZ-8CB01	CZ-8FB01	CZ-8CB01	CZ-8FB01	CZ-8FB02
NEW ON	9FC5	A8A8	AADE	AADE	88C8
NEW ON 9			A914	A914	860D
NEW ON 8			9C43	A6C1	837D
NEW ON 7			9B74	A587	7998
NEW ON 6			9AB9	A4CC	7812
NEW ON 5	NEW ON &H＊＊＊＊で指定したアドレス		9A5D	A3A8	7476
NEW ON 4			95F8	9F43	7327
NEW ON 3			92FA	9BB4	6DCC
NEW ON 2			8BF0	9449	63FF
NEW ON 1			8945	919E	6345
NEW ON 0			773D	7F96	5C62
テキストスタートを表わすポインタ	9D52	A635	74DB	7D34	0016
変数エリアの始まり(テキストエンド＋1)を表わすポインタ	9D46	A629	74CF	7D28	0075

ＩＯＣＳをＸ１用に作り替えて移植するソフトが計測技研※１やスガヤ無線※２から発売されたています。

※１】MZ－2000BASICシステムソフトウエア・コンバータ（定価3,800円）

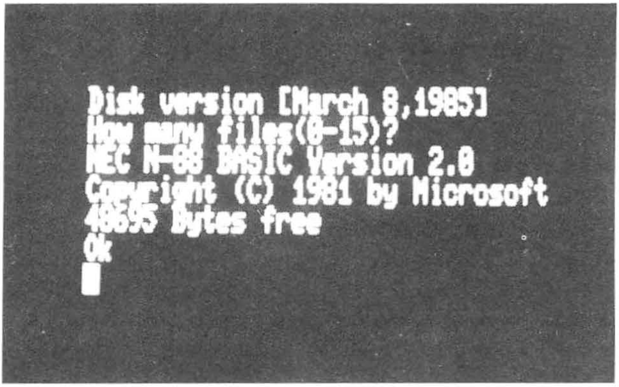
※２】漢字ベーシック（５インチ２Ｄ定価18,000円）

NECというのは伝統的？に、まとまったBIOSを作るのが嫌いらしく、表示ルーチンやカセット入出力ルーチンなどがBASIC ROM全体にバラまかれています。こういう作りかたをすると、例えばBASICにバグが見つかった場合などで一部を手直しすると、BIOSルーチンのエントリー・アドレスがずれてしまうことがあります。

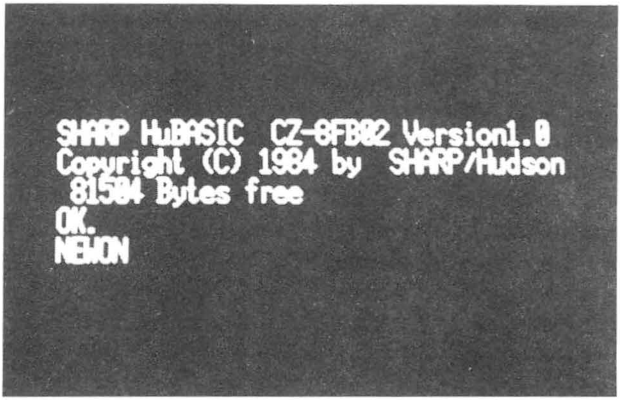
ＲＯＭ内ルーチンの絶対番地をコールするプログラムが動かない等の騒動を起こすのはNECの専売特許のようですが、N-60mBASICやN88-BASICなどの解析を進めていくと、どうも行き当たりばったりで設計しているとしか思えないことがあります。

ＦＭ-７やパソピア、ＭＢ-Ｓ１、ＩＢＭ-ＰＣはてはＭＳＸまでしっかりとしたBIOSを持っているというのに、よりによって業界トップのＮＥＣだけがBIOSという考えができていないというのは、どういうことでしょう。このまま行けば、〝BIOSを知らない子供達〟がたくさんできてしまいます。

それはパソコンの将来にとって決して良いことではありません。ＮＥＣは業界トップの責任と自覚を持ち、BASICの設計を根本から改めてもらわないと困ります!!



NEC N-88 BASIC のフリーエリア



ＸＩターボ漢字BASICのフリーエリア

ところでIOCSはRAMですから、自由に書きかえて、ブレークを禁止したりテープ・フォーマットやボーレートを変えたりすることができます。この部分は旧BASICとNEW BASICでは全く同じに作られていますので、今まで使っていたマシン語ツール（例えばマイクロラブのスクロール・ディスアセンブラーなど）やマシン語プログラムで、IOCSを使っているものは、NEW－BASICでもそのまま動作します。なお、ターボの漢字BASICはIOCSにあたる部分がROMになっているため、中身が全々ちがっています。また、書き換えもできません。

BASICのバージョンと各プログラムの注意点

ここで注意意しておかなければならないことがあります。これから出てくるサンプルの中にはＸ１の全BASICで共通して使えるものやそれぞれのBASICでみなちがっているものなどいろいろあるということです。

そこで、混乱をさけるために、次のように分類することにします。

㊎：このマークがついているサンプルは全BASICで共通に使えることを示します。

㊔：各BASICでちがうので、サ

ンプルにしているBASICと、表3をよく見比べてそれぞれのBASICに読みかえて使ってください。

IOCS　旧BASICとNEW BASICで共通して使えるサンプルです。ターボの漢字BASICでは使えませんが、旧BASICかNEW BASICを起動すればターボでも使えることを示します。

ターボ　ターボの漢字BASICと専用のサンプルです。

なお、㊔のサンプルは特にことわらない限りNEW－BASIC CZ－8FB01 V2.0のものです。

《カセットの入出力》

X１では、BASICプログラムのLOAD SAVEなどや、早送り、巻戻しなどのメカ制御をすべてIOCSを通して実行します。使い方はレジスタ(A、BC、DE、HLなど）に値をセットしてカセット入出力ルーチンのエントリー・ポイントをCALLするだけで良いのです。

カセットの入出力は、すべてソフトで行なっていて、PCのような8251（通信用IC）は用いていませんし、IOCSはすべてRAMですから少し書き換えるだけでテープ・フォーマットやボーレートなどが自由に変えられます。カセット入出力で重要なポイントを表4にまとめておきました。

テープ・フォーマットを変えたプロテクトは、後程詳しくお話しますが、カセット・ルーチンの使い方の一例をリスト3に載せておきます。

この使い方ですが、まずBASICを起動したあと適当なテープ(マシン語でもBASICでも良い）をセットして、リスト3を実行してください。この後、MON⏎でモニタに入り、D1480 149F⏎として1480Hから149FHまでのメモリの内容をのぞいてみましょう。(リスト4）

すると、ファイル・ネームやファイルの種類(BASICかマシン語かなど)、マシン語のスタート・アドレスや実行アドレスなどの情報を含んだ部分（FCB：ファイル・コントロール・ブロック）の内容を見ることができます。

これがカセットの入出力ルーチンの基本的な使い方です。カセット・ルーチンのより高度な使い方やFCBの構造については、後程詳しく見て行くことにします。

●BASICプログラムの内部構造

アドベンチャー・ゲームなどはLISTを取られてしまうと、それだけで結果が分かり興ざめてしまいます。そこで親切心からソフト・ハウスは、皆さんのためにプロテクトをかけてくれる！わけですが、見られないとなると見たくなるのが人情です。

場合によっては、アドベンチャーそのものよりプロテクトとの戦いの方が面白くなったりして……。(それでこそハッカーなのだ！）しかし、実際プロテクトをはずすとなると、BASICやBASICプログラムの構造を詳しく知っていなければなりません。

ここではまず、BASICプログラムの構造が一体どうなっているのか？ということから見て行くことにしましょう。また、BASICプログラムの構造を知ることから自ら新しい方法を見付けだすことがあります。

自分で作ったアドベンチャーにどぎついプロテクトをかけて、たくさんの人をアツ！といわせ、うならせ、悩ませてみるのも面白いのではないでしょうか。

まず、BASICを起動してリスト5を打ち込んでください。次に＊MON⏎でモニタに移りテキストのスタートをダンプしてください。テキストのスタート番地は各BASICやNEW ONの値によってそれぞれ異なってきますので、表3を参考にしてください。

ここではCZ－8FB01V2.0を例に取りますと、リスト6のようになります。

＊DAADE⏎

AADEHから最初の２バイトは、リンク・ポインタといい、１行の長さを示しています。00A0Hですから、

AADEH＋A0H－1＝AAE7Hまでが、ひとまとまりであることが分かります。

また行の終わりは必ず00Hとなっています。そうすると、アレ？

２バイトあると、１行の長さは65535バイトまで許されることになります。しかし、実際にBASICからは最大でも255バイトまでしか設定することができません。どうやら、このあたりにUNLIST化の秘密がありそうですね。

次の２バイトは行番号を表しています。行番号も16進数で表わされていますので、０～65535行まで可能です。ここに00Hを書き込むと行番号０ができます。

〈表 4〉 CZ-8(C)FB01のIOCSのカセット関係ルーチンと、パラメーターの位置(旧 BASIC、NEW-BASIC 共通)

IOCS

●エントリー・アドレス

FCBのセーブ	003B$_H$	実行中にエラーが出るとキャリーフラグが1になる	○HLレジスタにFCBが置いてあるアドレス、BCレジスタにFCBの長さ(0020H)をセットしてコール
データー部のセーブ	003E$_H$		○HLレジスタにデーターが置いてあるアドレス、BCレジスタにデーターの長さをセットしてコール
FCBのロード	0041$_H$		○HLレジスタにFCBバッファのアドレス、BCレジスタにFCBの長さ(0020H)をセットしてコール
データー部のロード	0044$_H$		○HLレジスタにデータを置くアドレス、BCレジスタにデーターの長さをセットしてコール
データー部のベリファイ	0047$_H$		○HLレジスタにデーターが置いてあるアドレス、BCレジスタにデーターの長さをセットしてコール
1バイトライト	0C8A$_H$		○Aレジスタにデーターをセットしてコール
1バイトリード	0CAA$_H$		○コールすると、Aレジスタに1バイト読み込んで帰る
カセットリーダー音書込	0CDF$_H$		
カセットのリーダー音待ち	0D20$_H$		DレジスタのBIT3が1ならFCB、0ならデータ部の待ち合わせ
信号の立ち上りを待ち合わせる	0D5C$_H$		
カセットへ0を1ビット書込	0D8A$_H$		
カセットへ1を1ビット書込	0DA4$_H$		
カセットのコマンド実行	0DEC$_H$		○Aレジスタにカセットコントロールコード、00$_H$～0A$_H$をセットしてコール

●パラメーター及びワークエリア

			通常	
チェックサムデータ	0EA1$_H$		00	
ギャップの数 WRITE時	0CE5$_H$、0CE6$_H$	テープ・フォーマット変更	14、14	この3つの値は必ず同じでなければならない
READ時ギャップ数	0D26$_H$		14	
0のH期間	0D94	ボーレート変更	20	
0のL期間	0D9D		18	
1のH期間	0DAF		44	
1のL期間	0DB8		3C	
サンプリングタイミング	0DC0		2E	



3年間の沈黙を破り、ＭＺからついにとんでもない化け物マシンが登場しました。

FM－16βよりも速いグラフィックス，4096色パレット、ディスクを使わない文節変換、漢字キャラジェネ，スムース・スクロール(ゼビウスが面白そう)、ＦＭ音源、ハードウェア・ウインドウ、フルロジック、モデムホン、メイン・メモリ最大512Ｋ、BASICインタプリタの大きさ約120Ｋなどスペック上では現在あるパソコン（オフコンも含めて）の中でＭＺ－2500を上回るものはないそうです。(ちなみにCPUはＺ－80Ｂ)

はじめて発表会でＭＺ－2500をみたときの感じは、この新製品の素晴らしさというよりも、私達が描いていた〝これからは16ビットそして32ビットへ〟というビジョンが完全に打ち砕かれたような気がしたということです。

ＰＣ－9801が現れたとき、漢字処理とグラフィックスの速さにため息をつき、さすがに16ビットと思ったものでしたが、いつの間にか私達は漢字Ｖ－RAMとGDC（グラフィック・ディスプレイ・コントローラ）を16ビットだと思い込まされていたようです。ところが、ＭＺ-2500にこうもあっさりと漢字Ｖ-RAMとGDCが積まれてしまうと、どんな16ビット・マシンもとたんに色あせて見えてきてしまうから不思議です。

私達はだまされていたのです。なぜなら、「現在のノイマン型コンピュータである限り（ボトルネックという障害があって）８ビットが32ビットになろうが飛躍的な性能は望めない。だから非ノイマン型チップを開発しているんだ」(日経紙)とは、NEC開発陣の言葉だからです。

最近のパソコンの多機能ぶりには目を見張るものがありますが、これらは全て周辺ＩＣの発達によってもたらされたものであるといって間違いはないでしょう。なぜならCPUはクロックが速くなった以外、これといって目立った変化がないからです。

時代の主役はいまやCPUから周辺ICへと移りつつあるのではないでしょうか!?現在でもＸ１ターボなどはCPU管理下のメモリよりもディスプレイ・コントローラ管理下のメモリのほうが多くなっています。将来、非ノイマン型チップがGDCに採用されるようになったり周辺ICが発達してゆくとCPUが主役の座を降りる時代がきっとやってくるでしょう。

そのときには、性能の差を論じるのにCPUが何ビットであるかは関係なくなるでしょう。その兆しがMZ—2500であるような気がしてなりません。

〈リスト2〉 NEW ONでけずられる命令とフリーエリア CZ-8FB02 漢字 BASIC

```
380 'NEWON...ALL COMMAND 使用可
390 'NEWON9..MIRROR$,KANJI$,DTL,RANDOMIZE,WAIT,KEYLIST,KLIST
400 '........KEYLLIST,KBUF,CANVAS,LAYER,TVPW,CHANNEL,VOL
410 'NEWON8..VERIFY,LOAD?,"CAS:",CMT,CMT関数,REW,FAST,EJECT,APSS
420 'NEWON7.."COM:",ON COM GOSUB,COM ON,COM OFF,COM STOP,ATTR$,RUN"?.Sys"
430 '........POSITION,PATTERN,CIRCLE@,SCROLL,STICK,STRIG,PUSH,POP
440 'NEWON6..CRT,ON KEY GOSUB,KEYnON,KEYnOFF,KEYnSTOP
450 '........ON TIME$="HH:MM/mm" GOSUB,TIME$ON,TIME$OFF,TIME$STOP:ON TIME$=""
460 'NEWON5..MKDIR,CHDIR,RMDIR,HDOFF,SET,NAME,FPOS
470 'NEWON4..MOUSE,MOUSE関数,HCOPY
480 'NEWON3..GET@,PUT@,CGPAT$,DEVI$,DEVO$,COPY,LPOUT,CONSOLE#
490 'NEWON2..LIST,LLIST,DELETE,RENUM,AUTO,EDIT,TRON,TROFF,SAVE,SEARCH
500 '........KILL,CONT,ｴﾗｰ ﾒｯｾｰｼﾞ
510 'NEWON1..PLAY,MUSIC,TEMPO,SOUND,CBLACK
520 'NEWON0..WINDOW,LINE,PSET,PRESET,CIRCLE,POLY,PAINT,SYMBOL,POINT
530 '........PRW,PALET,CLICK,NEWONn
```

システムディスクのStart up. Basをロードしてリストをとると書いてある

〈リスト3〉 カセットルーチンの使い方の例 IOCS

```
10 CLEAR &HF000
20 POKE  &HF000,&H01,&H20,&H00   'LD    BC,0020H
30 POKE  &HF003,&H21,&H80,&H14   'LD    HL,1480H
40 POKE  &HF006,&HCD,&H41,&H00   'CALL  0041H
50 POKE  &HF009,&H3E,&H01        'LD    A,01H
60 POKE  &HF00B,&HCD,&HEC,&H0D   'CALL  0DECH
70 POKE  &HF00E,&HC9             'RET
80 '
90 CALL  &HF000
```

〈リスト4〉 FCB部分のダンプリスト

IOCS

```
:1480=01 42 41 53 49 43 20 43 /.BASIC C
:1488=5A 38 43 42 30 31 20 20 /Z8CB01
:1490=20 20 10 9E 00 00 00 00 /   .ヽ....
:1498=82 B3 17 23 50 00 00 00 /▂ウ.#P...
```

行番号0の使い方はあとでお話します。

次の8FHはPRINT命令の中間コードです。命令や関数はBASICの内部では、そのまま〝PRINT〟として保存されているのではなく、こうした中間コードに置き換えられているのです。こうしておくと実行速度も上げられるし、メモリの節約にもなるからです。ここで漢字BASICの中間コード表を載せておきましたので、参考にしてください。（表5）

これはX1ターボのものですが、ない命令を除けばCZ-8(C)FB01とも共通です。BASICプログラムの終わりには、00Hが2つ入りますので、3個の00Hが続いています。リスト5の例ではAAE5Hがプログラムの終

わりであることが分かります。

こうして見るとBASICプログラムの構造は、PC、FM、X1ともほとんど同じだということがバックアップ活用テクニック①と見比べると良く分かると思います。ただ1つ、リンク・ポインタのところに各社の個性が出ているのが、とても面白いところです。

アドレス、つまりプログラムのテキスト・エンド＋2を示すポインタ（AB02H）が7D28Hに書かれています。これは、各BASICにより異なりますので、表3を参考にしてください。

〈リスト5〉

㊎

```
10 PRINT A
```

〈リスト6〉ダンプリスト

㊔

```
:AADE=08 00 0A 00 8F 20 41 [00] /....+ A.
:AAE6=[00 00] 00 00 4F FE 35 24 /....O 5$
```

テキスト・エンド　　END

サイクル・スチールとは

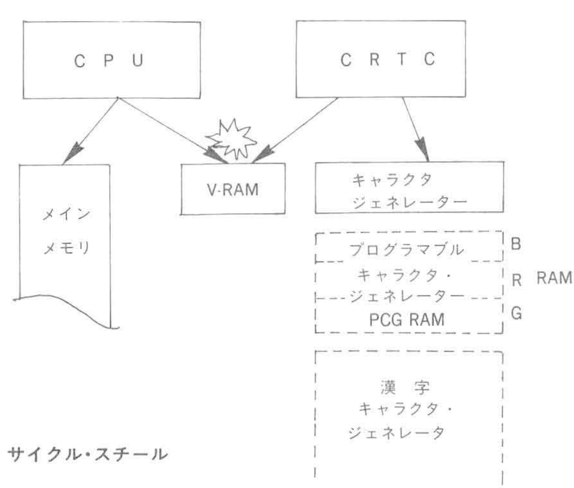
CPUは、コンピュータ内のすべてのROM・RAMを管理しているわけではありません。V-RAMやCG-ROMのようにCRTCが管理するメモリもあります。V-RAMなどは、両方から管理されているので、タイミングが悪いとバスの奪いあいをすることがあります。

これをさけるため、①定期的にCPUを止めてデータを転送する（SR以前のPC-8801：慢性的に遅い）②CRTCの休みの期間（垂直ブランキング）にアクセスする（PC-8001～II：表示がトロい）などの対策がしてありますが、これでは実行速度が落ちてしまいます。

そこでCPUがメモリをアクセスしないタイミング-エクゼキュート・サイクルを狙ってCRTがV-RAMをアクセスするのがサイクル・スチールです。（FM-77）

また、CPUクロックの前半でメモリのアクセスを終了してしまい、後半をCRTCに明け渡す方法もあります。（X1～ターボ）

どちらにしても見かけ上、同時にV-RAMをアクセスできるので、実行速度を落とさずにすみます。



サイクル・スチール

〈表5〉漢字Hu-BASIC中間コード表

L/H	0x	1x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx	FE+8x	FE+9x	FE+Ax	FE+Bx	FF+8x	FF+9x	FF+Ax	FE+Bx	FE+Cx	FF+Dx
x0	Endmark	&J 2	GOTO	LPRINT	POKE	DEFDBL	ERASE	CALC	TO	=<	WINDOW	LAYER	EFFECT	KSEN	INT	INP	CHR$	CVI	MEM$	JIS$
x1	0	&K 2	GOSUB	INPUT	OFF	DEFSTR	ERROR	FIELD	STEP	<=	PSET	CANVAS	GRAPH	LPOUT	ABS	CDBL	STR$	DEVF	SCRN$	KTN$
x2	1	整数2	GO	LINPUT	WHILE	DEF	ELSE	GET	THEN	=>	PRESET	CREV	MUSIC	SYMBOL	SIN	CSNG	HEX$		VARPTR	AKCNV$
x3	2		RUN	IF	WEND		CALL	PUT	USING	>=	COLOR	CFLASH	TEMPO	TERM	COS	CINT	OCT$	ERR	STRING$	KACNV$
x4	3		RETURN	DATA	REPEAT	LOAD	MON	SET	SUB	=	CIRCLE	CGEN	CURSOR	HDOFF	TAN		BIN$	ERL	TIME$	KLEN
x5	4	実数5	RESTORE	READ	UNTIL	SAVE	LOCATE	FILES	BASE	<	POLY	CSIZE	VERIFY	CBLACK	LOG	EOF	MKI$	CSRLIN	DAY$	KPOS
x6	5		RESUME	DIM		MERGE	SCREEN	LFILES	TAB	>	PAINT	EJECT	CLR	BCOLOR	EXP	FPOS	MKS$	STRPTR	DATE$	MOUSE
x7	6		LIST	REM		CHAIN	KEY	DEVICE	SPC	+		CSTOP	LIMIT	MKDIR	SQR	LOC	MKD$	DTL	FN	
x8	7	実数8	LLIST	END		CONSOLE		NAME	EQV	−	POSITION	FAST	KLIST	CHDIR	RND	LOF	SPACE$		USR	
x9	8		DELETE	STOP	TRON	WIDTH		KILL	IMP	MOD	PATTERN	REW	ASK	RMDIR	PEEK	POSFA	CGPAT$		CALC	
xA	9		RENUM	CONT	TROFF	OUT	LABEL	LSET	XOR	¥	HCOPY	APSS	KBUF	COPY	ATN	FAC	KANJI$	LEFT$		
xB	行No.		AUTO	CLS		SEARCH	RANDOMIZE	RSET	OR	/	PLAY	TVPW	CLICK		SGN	SUM	ASC	RIGHT$	ATTR$	
xC	アドレス		EDIT	CLEAR		WAIT	OPTION	INIT	AND	*	SOUND	CHANNEL	BOOT		FRAC	FRE	LEN	MID$	POINT	
xD	&O 2		FOR	ON		PAUSE	LINE	VDIM	NOT	^	BEEP	VOL	DEVI$		FIX	LPOS	VAL	INKEY$	CHARACTER$	
xE	&B 2		NEXT	LET	DEFINT	WRITE	OPEN	MAXFILES	><	(Ext.)	PRW	CRT	DEVO$		PAI	STICK	CVS	INSTR	CMT	
xF	&H 2		PRINT	NEW	DEFSNG	SWAP	CLOSE	COM	<>	(Func.)	PALET		KMODE		RAD	STRIG	CVD	HEXCHR$	MIRROR$	

PCGとは

Ｘ１のサンダーフォースは８色フルカラーで８方向スクール、その上やたらに速いのです。これはＰＣＧの機能をフルに使っているからです。ＰＣＧの機能はＭＳＸのスプライトと似ていますが、はるかに使いやすいのです。ゲームには不可欠な機能ですがＦＭ-７やＰＣ-8801(-ＳＲ）にはありません。

ＰＣＧは８×８ドットのマス目に８色フルカラーのグラフィックを設定し、これを画面内のどの位置にでも表示できるというものです。電源を入れると直ちに〝ＡＢＤ……〟という文字が使えるのはキャラクタ・ジェネレータというＲＯＭがあって、これが文字の形を覚えているからです。

これをＲＡＭにすれば自由に形を作る（プログラマブル）ことができます。〝ＡＢＣ……〟などのキャラクタと同じ速さで表示できますから、キャラクタの移動などはグラフィックを使うよりはるかに高速です。

また、バックとキャラクタが重なった部分の処理は全てハードが行なってくれる（プライオリティという）ので、めんどうくさい処理がいりません。

Ｘ１はＰＣＧ ＲＡＭが６Ｋバイトあり、256個のキャラクタを設定できます。ＰＣＧを設定するにはデータをいったんメイン・メモリに取り組み、Ｉ/Ｏポートを通してデータを転送します。

チェックサム付きマシン語入力プログラム

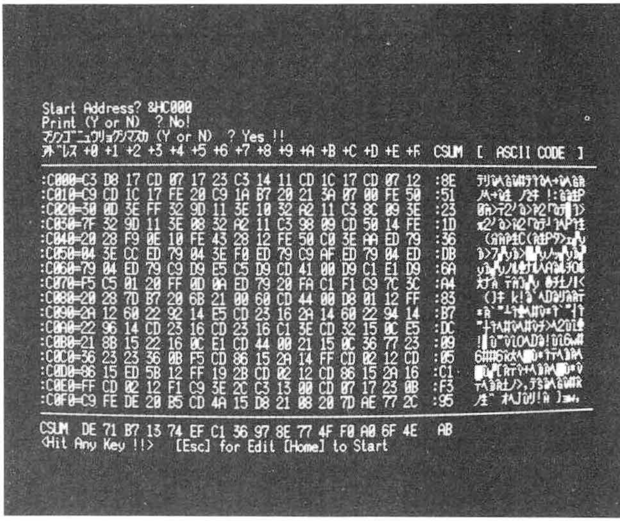
●使い方

プログラム・リストを打ち込みRUNするとスタート・アドレスを聞いてきますので、16進数で入力してください。すると、そのアドレスから256バイト分が表示されます。

マシン語を入力する時は、[ESC]キーを押してエディット・モードに入ってください。例に従ってワクの中だけを打ち込んでください。

全部打ち込んだら、[HOME]：↵とすると再びダンプが表示されますから、この段階でチェックサムを見比べます。全部OKなら、スペースを押すと次に進みます。

本プログラムで見たダンプ・リスト




```
ｱﾄﾞﾚｽ +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  CSUM
:E800=18 0E 18 0F 3E C3 21 00 E7 32 E4 F8 22 E5 F8 C9  :2C   ....>ﾃ!..2..".ﾉ
:E810=CD 04 E8 21 1A E8 22 DF F8 C9 3A 8F F8 4F 3A A2  :8A   ﾍ..!..ﾟ".ﾉ:..O:｢
:E820=F8 B1 07 D8 21 99 F8 CD 27 39 CD 33 E8 21 94 F8  :FC   .ｱ.ﾘ!..ﾍ'9ﾍ3.!..
:E830=C3 27 39 3A 71 FB 6F 3A E7 FA 85 6F 26 00 29 29  :BF   ﾃ'9:q.o:...o&.))
:E840=29 11 6D FC 3A 6E FB B7 CA BF E8 E5 D5 E5 3A A7  :EE   ).m.:n.ｷﾊｿ..ﾕ.:ｧ
:E850=F8 CD AA 6B E1 3A 71 FB 47 0E 00 C5 CD 6A E8 3A  :D4   .ﾍｪk.:q.G..ﾅﾍj.:
:E860=A8 F8 CD AA 6B C1 0E 01 D1 E1 78 B7 28 1D C5 CD  :0A   ｨ.ﾍｪkﾁ..ﾑ.xｷ(.ﾅﾍ
:E870=B5 6B EB C1 7E CD 99 30 38 15 05 23 E5 57 5E C5  :B4   ｵk.ﾁ~ﾍ.08..#.W^ﾅ
:E880=CD 81 2F CD DB E8 C1 E1 23 10 E9 CD 0F 38 C9 E5  :8D   ﾍ./ﾍﾛ.ﾁ.#..ﾍ.8ﾉ.
:E890=C5 06 14 CD 98 E8 18 EE 5F AF 57 C5 CD B2 E9 C1  :85   ﾅ..ﾍ...._ｯWﾅﾍｲ.ﾁ
:E8A0=21 7F FD ED 78 77 23 0C 0C 79 FE 10 38 F5 0E 08  :7E   !...xw#..y..8...
:E8B0=21 7F FD 06 08 CD B0 F8 CD 83 39 0D 20 F2 C9 7C  :0D   !....ﾍ.ﾍ.9. .ﾉ|
:E8C0=B5 28 15 CD B5 6B 3A 71 FB 47 1A D5 C5 01 00 14  :95   ｵ(.ﾍｵk:q.G.ﾕﾅ...
:E8D0=CD 98 E8 C1 D1 13 10 F2 C3 8B E8 21 C0 E9 7E 23  :95   ﾍ..ﾁﾑ...ﾃ..!ﾀ.~#
:E8E0=B7 28 0A BB 20 04 7E BA 28 17 23 18 F1 7E 23 B7  :C3   ｷ(.ｻ .~ｺ(.#..~#ｷ
:E8F0=28 3B BB 20 04 7E BA 28 1B C5 01 21 00 09 C1 18  :86   (;ｻ .~ｺ(.ﾅ.!..ﾁ.

CSUM  53 D3 0E 0A 8B 89 EB E1 63 5A 72 8B 81 5A 1F 2F   01
```

ココまで来たらリターンを押す

[HOME]・[RET]で正しいチェックサムが出力される

チェックサム付きマシン語入力ダンプ・プログラムの説明

```
10 DEFINT A-Z,DIM D(20):ONERROR GOTO560
20 WIDTH 80:CLS
30 INPUT "Start Address? &H",SA$
40 PRINT "Print (Y or N)  ? ";:K$=INPUT$(1)
50 IF K$<>"y" AND K$<>"Y" THEN P=0:PRINT"No!" ELSE P=-1:PRINT"Yes !!"
60 PRINT "ﾏｼﾝｺﾞﾆｭｳﾘｮｸｼﾏｽｶ (Y or N)  ? ";:K$=INPUT$(1)
70 IF K$<>"y" AND K$<>"Y" THEN PRINT"No!" ELSE PRINT"Yes !!":CLEAR VAL("&H"+SA$)-1000:CLEAR VAL("&H"+SA$)
80 SA=VAL("&H"+LEFT$(RIGHT$("0"+SA$,4),3)+"0")
90 '
100 IF P THEN LPRINT:LPRINT
110 FOR A=0TO 16:D(A)=0:NEXT
120 P1$="ｱﾄﾞﾚｽ +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  CSUM  [  ASCII CODE  ]"
130 PRINT P1$:PRINT STRING$(79,"-")
140 IF P THEN LPRINT P1$:LPRINT STRING$(79,"-")
150 FOR A=SA TO SA+255 STEP 16
160 CS=0:C=0:P2$=":"+RIGHT$("000"+HEX$(A),4)+"="
170 PRINT P2$;:IF P THEN LPRINT P2$;
180 P3$=""
190 M$=MEM$(A,16)
200 FOR B=1 TO 16
210 E=ASC(MID$(M$,B,1)):CS=CS+E:D(C)=D(C)+E
220 P3$=P3$+RIGHT$("0"+HEX$(E),2)+" ":C=C+1
230 NEXT
240 PRINT P3$;:IF P THEN LPRINT P3$;
250 D(16)=(D(16)+CS AND 255):P4$=" :"+RIGHT$("0"+HEX$(CS),2)+"   "
260 PRINT P4$;:IF P THEN LPRINT P4$;
270 PRINT#0,M$:IF P ELSE 350
280 O$=""
290 FOR B=1 TO 16
300 N$=MID$(M$,B,1)
310 IF N$<" " OR (N$>"ｧﾋﾞ" AND N$<CHR$(&HA0)) OR N$>"°" THEN N$="."
320 O$=O$+N$
330 NEXT
340 LPRINT O$
350 NEXT
360 PRINT STRING$(79,"-"):PRINT "CSUM  ";
370 IF P THEN LPRINT STRING$(79,"-"):LPRINT "CSUM  ";
380 FOR A=0 TO 15:P6$=RIGHT$("0"+HEX$(D(A)),2)+" "
390 PRINT P6$;:IF P THEN LPRINT P6$;
400 NEXT
410 P7$="  "+RIGHT$("0"+HEX$(D(A)),2):PRINT P7$:IF P THEN LPRINT P7$:LPRINT
420 PRINT "<Hit Any Key !!>   [Esc] for Edit [Home] to Start";CHR$(11):GOSUB600
430 IF INKEY$(0)="" THEN 430 ELSE K$=INKEY$:IF K$=CHR$(&H1B) THEN 470 ELSE IF K$=CHR$(&HB) THEN 20
440 SA=SA+256
450 LOCATE 0,3:PRINT CHR$(&H1A);
460 GOSUB 600:GOTO 100
470 'Edit
480 LOCATE 0,23:PRINT "< Edit Mode !! >  [Home]|[CR] for RET.";SPC(11);
490 LOCATE 6,5
500 LOCATE 6,CSRLIN:LINEINPUT "",A$
510 IF MID$(A$,1,1)=":" AND MID$(A$,6,1)="=" ELSE 450
520 A=VAL("&H"+MID$(A$,2,4)):B=7
530 IF MID$(A$,B,1)=" ":B=B+1:GOTO 530
540 IF MID$(A$,B,1)=":":GOTO 500
550 POKE A,VAL("&H"+MID$(A$,B,2)):A=A+1:B=B+2:GOTO 530
560 'Error
570 IF ERR<>5 AND ERR<>7 THEN ONERROR GOTO0
580 CFLASH 1:PRINT "BASIC ｶﾞｺﾜﾚﾏｽ !!":CFLASH
590 FOR I=1 TO 100:GOSUB 600:NEXT:RESUME 20
600 BEEP 1:FOR T=1 TO 50:NEXT:BEEP 0:RETURN
610 '==========================
620 '   ASCII Dump Program V.3
630 '   Copyright 1985.9.4
640 '           Saburou-Tama
650 '==========================
```

チェックサム付きマシン語入力ダンプ・プログラム

```
ｱﾄﾞﾚｽ +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  CSUM  [  ASCII CODE  ]
--------------------------------------------------------------------------------
:FF00=01 38 30 30 31 43 6F 6E 76 65 72 74 65 72 20 20  :C2   .8001Converter
:FF10=20 20 3C 00 C0 FF 00 00 83 15 07 04 26 00 00 00  :04     <.9.......&...
:FF20=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF30=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF40=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF50=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF60=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF70=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF80=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF90=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FFA0=DA 2D 10 3E 16 32 26 0D 22 1F 10 2E 22 36 01 21  :C9   ﾚ-.>.2&."..."6.!
:FFB0=A0 14 CD 9E 0B 3E 14 32 26 0D D8 E9 00 DA F5 00  :71    .ﾍ..>.2&.ﾘ..ﾚ..
:FFC0=B6 B2 BE B7 BA DE B8 DB B3 BB CF C3 DE BD A1 00  :44   ｶｲｾｷｺﾞｸﾛｳｻﾏﾃﾞｽ｡.
:FFD0=38 09 21 00 F7 01 D9 08 CD 9E 0B DA 97 0B E5 C3  :D5   8.!...ﾙ.ﾍ..ﾚ...ﾃ
:FFE0=EA 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :F7   ................
:FFF0=90 FF A0 FF B0 FF C0 FF D0 FF E0 FF BD FF 44 00  :4A   .. .-.ﾀ.ﾐ...ｽ.D.
--------------------------------------------------------------------------------
CSUM  03 60 C8 C2 73 90 FA 8F 91 FE 1B 2B DF 49 E0 04   5A
```

"N-BASIC→X1テキストコンバーター"
ナニ／カイセキゴクロウサマだと!?解いてやる／解いてやるぞテメエなんか…／

```
ｱﾄﾞﾚｽ +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  CSUM  [  ASCII CODE  ]
--------------------------------------------------------------------------------
:FF00=01 4E 2D 42 41 53 49 43 BA DD CA DE 2D C0 20 20  :4A   .N-BASICｺﾝﾊﾞｰﾀ
:FF10=20 D2 73 00 90 FF 90 FF 84 51 21 03 21 40 00 10  :ED    ﾒs......Q!.!@..
:FF20=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF30=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF40=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF50=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF60=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF70=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF80=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00   ................
:FF90=CD BE FF 21 00 90 01 90 6F CD 44 00 D2 03 90 FE  :AF   ﾍｾ.!....oﾍD.ﾒ...
:FFA0=DA F5 00 57 D9 E1 D1 C1 54 ED B0 21 3E 1D 22 F8  :F9   ﾚ..Wﾙ.ﾑﾁ[.-!>.".
:FFB0=00 E1 2B 25 35 35 4E F1 E8 C0 D1 ED 42 C1 D9 E5  :01   ..+%55N..ﾀﾑ.Bﾁﾙ.
:FFC0=D9 ED 42 06 1E 7C 87 2F ED 79 77 CB 1D 2C 99 32  :1A   ﾙ.B..|./.ywﾋ.,.2
:FFD0=FA 00 E5 D9 E0 C4 41 00 18 6A DA C9 CC DF DB C3  :EB   ...ﾙ.ﾄA..jｺﾉﾌﾟﾛﾃ
:FFE0=B8 C4 A6 C4 B6 DA C0 B6 C0 CA 20 B5 C0 D6 D8 B8  :71   ｸﾄｦﾄｶﾚﾀｶﾀﾊ ｵﾀﾖﾘｸ
:FFF0=C0 DE BB B2 58 10 FF 61 00 0D 44 00 A0 FF C3 FF  :85   ﾀﾞｻｲX..a..D. .ﾃ.
--------------------------------------------------------------------------------
CSUM  13 43 52 34 EB 22 80 CA AE 62 45 38 E9 C1 BA B7   DB
```

"N-BASIC X1 システムソフトウェアコンバーター"プロテクトを解かれた方はお便りください。
BASIC HOUSEのプロテクトは超一流だ!!

BASICダンプ・リスト

パート①と同じBASICダンプ・リストのX1版を紹介します。RUN 60000とするとどこの番地からダンプするのかを聞いてきます。BASICのプログラムの始まりは表3を参考にしてください。

アドレスを入力するとダンプしはじめます。スペース・キーで一時停止、ESCキーでエディット・モードに入ります。直接カーソルキーを持っていて書き換え、⏎キーを押してください。

```
60000 WIDTH 80
60010 PRINT CHR$(&H1A)
60020 INPUT "Start Address? =&H",A$
60030 PRINT "Print (Y or N) ? ";
60040 K$=INPUT$(1)
60050 L=K$="y" OR K$="Y"
60060 IF L :P=-1:PRINT "Yes !!"
60070 IF NOT L :P=0:PRINT "No!"
60080 A=VAL ("&H"+A$)
60090 A$=":"+HEX$(A)+"=":PRINT A$;:IF P :LPRINT A$;
60100 BADR=A
60110 PONT=PEEK(A)+256*PEEK(A+1):IF PONT ELSE END
60120 FOR I=1 TO PONT:A$=RIGHT$("00"+HEX$(PEEK(BADR)),2)+" ":PRINT A$;:IF P :LPRINT A$;
60130 BADR=BADR+1:NEXT
60140 PRINT:PRINT "$       ";:IF P :LPRINT:LPRINT "$       ";
60150 BADR=A
60160 FOR I=1 TO PONT:A$=CHR$(PEEK(BADR)):B$=A$+"  ":PRINT#0,B$;:IF P :IF A$〈" " OR (A$〉CHR$(&H7F) AND A$〈
CHR$(&HA0)) OR A$〉"°" :LPRINT ".  "; ELSE LPRINT B$;
60170 BADR=BADR+1:NEXT
60180 PRINT:IF P :LPRINT
60190 A=A+PONT:A$=INKEY$
60200 IF A$="" :GOTO 60090 ELSE IF INKEY$(0)=" " :GOTO 60200
60210 KEY0,"":A$=INKEY$(1):IF A$=CHR$(&H1B) ELSE 60090
60220 IF A$=CHR$(&H1B) ELSE 60090
60230 PRINT:LINEINPUT A$
60240 IF MID$(A$,1,1)=":" AND MID$(A$,6,1)="=" THEN 60260
60250 IF MID$(A$,1,1)="$" THEN 60010
60260 A=VAL("&H"+MID$(A$,2,4))
60270 B=7
60280 IF MID$(A$,B,1)=" ":B=B+1:GOTO 60280
60290 IF MID$(A$,B,1)="":GOTO 60230
60300 POKE A,VAL("&H"+MID$(A$,B,2))
60310 A=A+1:B=B+2:GOTO 60280
60320 '***************************
60330 '    BASIC Dump & Edit
60340 '  Copyright by S.Tama
60350 '***************************
60360

:AADE=0A 00 60 EA B9 20 12 50 00 00
$        .  .  .  ｹ  .  P  .  .
:AAE8=0E 00 6A EA 8F 20 FF A0 28 0F 1A 00 29 00
$        .  .  j  .  .  .  .  .  .  .  .  )
:AAF6=1E 00 74 EA 91 20 22 53 74 61 72 74 20 41 64 64 72 65 73 73 3F 20 3D 26 48 22 2C 41 24 00
$        .  .  t  .  .     "  S  t  a  r  t     A  d  d  r  e  s  s  ?     =  &  H  "  ,  A  $  .
:AB14=1B 00 7E EA 8F 20 22 50 72 69 6E 74 20 28 59 20 6F 72 20 4E 29 20 3F 20 22 3B 00
$        .  .  ‾  .  .     "  P  r  i  n  t     (  Y     o  r     N  )     ?     "  ;  .
:AB2F=0D 00 88 EA 4B 24 F4 91 24 28 02 29 00
$        .  .  .  .  K  $  .  .  $  (  .  )  .
:AB3C=16 00 92 EA 4C F4 4B 24 F4 22 79 22 20 EB 20 4B 24 F4 22 59 22 00
$        .  .  .  .  L  .  K  $  .  "  y  "     .     K  $  .  "  Y  "  .
:AB52=19 00 9C EA 93 20 4C 20 3A 50 F4 F8 02 3A 8F 20 22 59 65 73 20 21 21 22 00
$        .  .  .  .  .     L     :  P  .  .  .  :  .     "  Y  e  s     !  !  "  .
:AB6B=17 00 A6 EA 93 20 ED 20 4C 20 3A 50 F4 01 3A 8F 20 22 4E 6F 21 22 00
$        .  .  ﾗ  .  .     .     L     :  P  .  .  :  .     "  N  o  !  "  .
```

1行のはじまりのアドレス

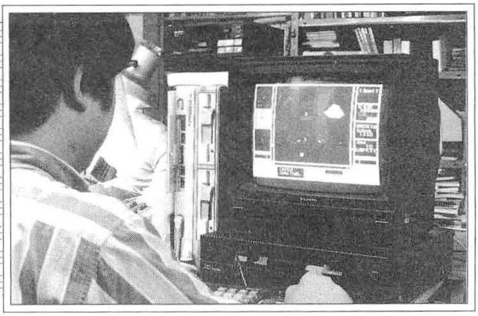
次の行をあらわすリンク・ポインタ AADE$_H$ + 000A$_H$ = AAE8$_H$

1行の終わり（00H）

行番号(60000)

プログラムにイタズラしてみよう

BASICプログラムをイジル



BASICのプログラムの中身が分かってくると、モニタから直接書き換えることで、BASICではできない、いろいろなことができるようになります。

●GOTO(GOSUB)文の秘密

ロードする時には普通にロードできて、セーブすると中身がメチャメチャになってしまうイタズラを紹介しましょう。

GOTO文やGOSUB文はBASICの中で多用されますので、これがあまり遅いとBASIC全体のスピードが落ちてしまいます。そこでGOTO文やGOSUB文には次のような工夫がされています。

まずリスト7を打ち込んでください。これをモニタからダンプしたものが、リスト8です。次にR⏎でBASICに戻り、一度RUNした後に、再び同じところをダンプしたものがリスト9です。

リスト8ではAAE2HがGOTO文の中間コード、AAE5Hから2バイトが行番号で16進数に変換されて下位、上位の順で入っています。これがRUNした後ではリスト9のように行番号が置かれているアドレスに書き換えられているのです。

GOTO文やGOSUB文は一度目の実行で行番号をサーチし、行番号のところをアドレスに書き換えますので、2度目の実行からは直接目的の行に飛んで行くことができるわけです。GOTO文の飛び先が行番号なのかアドレスなのかを知るにはAAE4Hを見ます。

ここが0BHなら行番号、0CHならアドレスというわけです。

以上のことが分かってくると、これを利用して次のようなイタズラができます。

リスト10を入れて必ず一度RUNしてください。リスト11はダンプ・リストです。モニタから、

：AB09H＝3C ⇨ 1E

に書き換えてBASICに戻りLISTを取ると、(リスト12)アレッ!?10行のGOTO 30はどちらを指しているのでしょうか。実

〈リスト7〉

```
10 GOTO 30
20 GOTO 30
30 END
```

〈リスト8〉

```
㊔
:AADE=0A 00 0A 00 80 20 [0B] [1E] /....※..
:AAE6=[00] 00 0A 00 14 00 80 20 /......※
:AAEE=0B 1E 00 00 06 00 1E 00 /........
:AAF6=98 00 00 00 00 00 4F 7C /※....0|
```

〈リスト9〉

```
㊔
:AADE=0A 00 0A 00 80 20 [0C] [F2] /....※.
:AAE6=[AA] 00 0A 00 14 00 80 20 /ｴ.....※.
:AAEE=0B 1E 00 00 06 00 1E 00 /........
:AAF6=98 00 00 00 00 00 4F 7C /※....0|
```

〈リスト10〉

```
10 GOTO 60
20 REM
30 BEEP
40 PRINT "ｹｹｹ"
50 END
60 PRINT "ﾊﾊﾊ"
```

〈リスト12〉

```
10 GOTO 30
20 REM
30 BEEP
40 PRINT "ｹｹｹ"
50 END
30 PRINT "ﾊﾊﾊ"
OK
RUN
ﾊﾊﾊ
OK
```

行してみると、どうやら後の方の30行を指しているようです。

メモリにイタズラしたままのプログラムをSAVEするにはモニタから次のように書きます。

＊ M1083 ⏎　　IOCS

：1083＝01⇨02

次にテキストのスタートとエンドを調べます。テキストのスタートとエンドを示すポインタはそれぞれのBASICで違いますので、P.102の表3を参考にし調べてください。

リスト11の例ではテキストはAADEH～AB14Hまでであることが分かります。これを次のようにしてセーブします。

＊ SAADE　AB14　0000：TEST ⏎

このプログラムをLOADしてRUNすると、後の方の30行が実行されます。このプログラムをBASICから再びSAVEして

〈リスト11〉

```
:AADE=0A 00 0A 00 80 20 0B 3C /....※.<
:AAE6=00 00 06 00 14 00 97 00 /......※
:AAEE=07 00 1E 00 FE 8D 00 0C /....※..
:AAF6=00 28 00 8F 20 22 B9 B9 /.(.※"ｹｹ
:AAFE=B9 22 00 06 00 32 00 98 /ｹ"...2.
:AB06=00 0C 00 3C 00 8F 20 22 /...<.※"
:AB0E=CA CA CA 22 00 00 00 00 /ﾊﾊﾊ"....
                        テキスト・エンド
```

LOADすると、今度は前の方の30行が実行されてしまうのです。

これなら皆さんの作ったゲームにこのイタズラをして、友達に「面白かったらコピーしてもいいよ」なんていって、友達を混乱させてみるのも面白いのではないでしょうか。

一度セーブするとストーリーが変わるアドベンチャー・ゲームなんていうのもできそうです。

ただし、このイタズラはBASICの種類と、NEW ONの値が同じでないと通じません。また、LOADしたらすぐにRENUMをかけられると破られてしまうのです。

SHARP/Hû-BASICはバカ正直ですね。RENUMでも破られない方法はないのでしょうか。

●アレッ、サブルーチンが無い

GOTO（GOSUB）文の飛び先に、直接アドレスが書き込めることを利用すると、次のようなことができます。

リスト13を打ち込んでモニタに移ってください。（リスト13）モニタから次のように書き換えて、

：AAE4＝0B ⇨ 0C

（GOTO文の飛び先はアドレス）

：AAE5＝1E ⇨ EE

〈リスト13〉

```
10 GOSUB 30
20 END
30 PRINT "ｱﾝﾀﾊｴﾗｲ!"
40 RETURN
```

〈リスト14〉

```
:AADE=0A 00 0A 00 81 20 0B 1E /....※..
:AAE6=00 00 06 00 14 00 98 00 /......※
:AAEE=11 00 1E 00 8F 20 22 B1 /....※"ｱ
:AAF6=DD C0 CA B4 D7 B2 21 22 /ﾝﾀﾊｴﾗｲ!"
:AAFE=00 06 00 28 00 84 00 00 /...(.※.
:AB06=00 00 00 4F 7C 36 3C 00 /...O|6<.
```

：AAE6＝00⇨AA
（アドレス設定）
：AAE8＝06⇨1D
（リンク・ポインタの書き換え）

0006H＋0011H＋0006H
20行のリンク・ポインタ　30行の・ポインタ　40行のリンク・ポインタ

＝001DH

BASICに戻りLISTをとると、どうでしょう。30行と40行が消えてしまいました。（リスト15）しかし、RUNすると実行するとされますので、本当に消えてしまったわけではありません。

〈リスト15〉

```
10 GOSUB 30
20 END
OK
RUN
アンタハエライ!
OK
```

リンク・ポインタのいじり方によっては、もっと大きな範囲をUNLIST化することもできます。これをセーブするにはモニタからセーブしてください。この方法では、RENUMをとってもリストは表示しません。

●NEWしたプログラムを復活させる方法

せっかく作った大切なプログラムをうっかりNEWしてしまった!!しかし、あわてることはありません。というプログラム復活法をPCやFMでは良く見かけるのですが、どうゆうわけかX1に限ってあまり紹介されていないようです。

これはX1のBASICが、プログラムをブレークし止めて修正しても、それまで実行した変数や配列の値を保存しているということにも関係があるのです。

X1はプログラム・エリアの始まりと終わりを示すポインタ以外にも変数エリアの終わりを示すポインタ等があり、これらを全て直さないと、プログラムの復活がうまくいかないのです。

これには、モニタからセーブする方法を使えば復活できます。

【手順1】NEWしても元のプログラムはそのままメモリ上に残っていて、一番はじめのリンク・ポインタと行番号がゼロにされてしまうだけで次の行先頭（リンク・ポインタ）の位置を捜すと1行の長さが分かるので、これを先頭のゼロにされた部分に書き込む。同時に行番号も修正する。（ここまではFM、PCと同じ）

【手順2】GOTO（GOSUB）文のアドレス⇨行番号に戻すルーチンを実行する。各BASICでそれぞれ違うので、表6を参考。

例えば、CZ-8FB01V2.0では、
＊G2C0C ⏎

【手順3】テキスト（プログラム）の最期の部分（ゼロが3つ並んでいる）を見付ける。モニタのFコマンドを使うとすぐ見付かる。
＊FAADE FEFF 00 00 00 ⏎
：××××＝00　00　00/・・・
（××××＋2がプログラム・エリアの終わり）㊔

【手順4】モニタからBASICとしてSAVEする。　IOCS 注
＊M1083＝01⇨02
＊SAADE　テキスト・エンド
0000：file name ⏎

【手順5】BASICに戻り、NEWしてLOADで無事復活。

注：ターボはモニタがROMなので書き換えができません。そこで一担、マシン語としてセーブしてしまい、後ほど紹介するバックアップ・ツールによって属性を書き換えます。

●BASICの内部を書き換えたプロテクト

X1のBASICはRAMですから自由に書き換えができます。ですからブレーク禁止ひとつを取り上げてみても、

1. IOCSのキー入力ルーチンの一部をこわす。
2. ブレーク・チェックのジャンプ・テーブルをこわす。

〈表6〉 GOTO GOSUB文のアドレス→行番号ルーチン

旧BASIC	CZ8CB01V1.0	2FC1
	CZ8FB01V1.0	2FEF
NEW BASIC	CZ-8CB01V2.0	2C0C
	CZ-8FB01V2.0	2C0C
漢字BASIC	CZ-8FB02	08BA

3．ブレーク・チェックのルーチンそのものをこわす。

などがあり、さまざまです。（リスト16）

クリーン・コンピュータというのは、プロテクトをかける側にとっては、まさに天国。はずす側にとっては地獄です。

UNLIST、例えばLISTをしてもRUNしてしまうようにする場合には、コマンドのジャンプ・テーブルと呼ばれている部分を書き換えます。

表7は各BASICのコマンドのジャンプ・テーブルのアドレスとジャンプ先のアドレスです。

例えば、CZ-8FB01V2.0でLISTを使えなくするには、

POKE &H2A11, &H78, &H17

とすればLIST⏎とするだけでRUNしてしまいます。同じようにしてLLISTやAUTO、EDITを書き換えればUNLISTになるのです。

〈リスト16〉

```
1   0047                    9 ;
    0047                   10 LOAD?2
    0047 C3 AE 0B          11                JP      VERIFY
    004A                   12 ;
    004A     ジャンプ先を変える 13 BRKCHK
    004A C3 [33 03]        14                JP      BRKCK
    004D                   15 ;
    004D C3 88 09          16                JP      WITH48
    0050                   17
-------------------------------------------------------------
2   0330                    7
    0330                    8 ;
    0330 3E 1A ── 3Eにする   9          LD    A,1AH
    0332 [DB] 01           10          IN    A,(01H)
    0334 E6 01             11          AND   01H
    0336 C0                12          RET   NZ
    0337 3E E6             13          LD    A,0E6H
    0339 CD FE 0D          14          CALL  COMOUT
    033C CD 49 0B          15          CALL  IN49SB
    033F CD 49 0B          16          CALL  IN49SB
    0342 FB                17          EI
    0343 FE 03             18          CP    3
    0345 C9                19          RET
    0346                   20 ;
-------------------------------------------------------------
3   03DC                    7 ;
    03DC 20 DB             8          JR    NZ,NTFUNC
    03DE        ── 3Eにする  9 BKSTBF
    03DE [32] 36 00        10          LD    (BRKBUF),A
    03E1 18 D6             11          JR    NTFUNC
    03E3                   12 ;
    03E3                   13
```

〈表7〉 BASICコマンドのジャンプ・テーブル・アドレスとジャンプ先アドレス

	CZ-8CB01 V1.0		CZ-8FB01 V1.0		CZ-8CB01 V2.0		CZ-8FB01 V2.0		CZ-8FB02	
	ジャンプテーブル位置①	処理アドレス②	①	②	①	②	③	②	①	②
RUN	2CE5H	1795H	2D13H	17B0H	2A09H	1778H	2A09H	1778H	5551H	035DH
LIST	2CEDH	68F7H	2D1BH	61C3H	2A11H	8E49H	2A11H	96A2H	5559H	68A6H
LLIST	2CEFH	68F2H	2D1DH	61BEH	2A13H	8E44H	2A13H	969DH	555BH	689EH
RENUM	2CF3H	2F09H	2D21H	2F37H	2A17H	8CCBH	2A17H	9524H	555FH	6715H
AUTO	2CF5H	212FH	2D23H	2150H	2A19H	8C12H	2A19H	946BH	5561H	647BH
EDIT	2CF7H	30A7H	2D25H	30D5H	2A1BH	8C0AH	2A1BH	9463H	5563H	64FBH
SAVE(M)	2D49H	6C04H	2D77H	64D4H	2A6DH	8F45H	2A6DH	979EH	55B5H	69FAB
SEARCH	2D55H	6871H	2D83H	613DH	2A79H	8DD6H	2A79H	962FH	55C1H	6820H
MON	2D67H	0FE2H	2D95H	0FE2H	2A8BH	0FE2H	2A8BH	0FE2H	55D3H	107EH

●NEWONによるUNLIST

NEW-BASICやターボの漢字BASICではNEWON nによって不要な命令を削り、その分フリー・エリアを増やすことができます。

NEWON 2とするとLISTやEDIT、SAVEまで削り取られてしまいます。(表8)

しかし、NEWON 1でもLINEや高速ペイントなどの必要最少限の機能は残るようになっています。何だかアドベンチャー・ゲームのために用意されたような命令ですね。

ただし、NEWON 1ではプログラムを作ることは不可能に近いと思いませんか。NEW ON 3でプログラムを作ってNEWON 1で走らせても良いのですが、NEWON 1とNEWON 3ではフリー・エリアが2.5Kバイトも違うので、この分がとてももったいないわけで、何とかフリー・エリアをフルに使う方法はないのでしょうか。

実はあるのです。MERGE命令だけは生きていますので、プログラムを前半と後半に分けて作っておき、どちらかをディスクかテープにアスキー形式でセーブするのです。

SAVE "TEST", A ↵

もう半分をメモリにロードした後、

MARGE "TEST" ↵

とすれば良いのです。できあがったプログラムはBASICからセーブできません。しかたがないのでモニタから、先ほど紹介した方法でセーブしてください。このようにして作ったBASICプログラムはNEWON 1以外でロードしてもOUT of memoryのエラーが出てロードできません。ターボの場合、とりあえずマシン語としてテープにセーブしておいてください。

また、ターボの漢字BASICにはPオプションがあります。

SAVE "TEST"、P ↵

とすると、このプログラムをロードしたとたんオートランしてしまい、ブレークが効かなくなってしまいます。一方、UNLISTをはずすにはまず、

1．全てのディスクとテープを抜く。

2．SHIFT＋BREAKしながらIPL（赤）ボタンを押す。

3．MAKE YOUR DEVICE READY

F：FLOPPY
R：ROM
C：CMT

の表示が出たら、SHIFT＋#を押す。(隠しコマンド)

4．OKが出たらMONでモニタに移り、NEWしたプログラムを復活させる方法でBASICプログラムの範囲を探り、その部分をとりあえずマシン語としてテープにセーブします。

5．後のところで紹介しているバックアップ・ツールで属性をBASICに書き直します。

●IPL起動のBASICプログラム

プログラムを実行するのに、BASICをロードしてそれからプログラムをロードしてと、いちいち面倒くさいと思いませんか。

こんな時、クリーン・コンピュータならではのBASICとプログラムをまとめてセーブしてしまう！という必殺技法があるのです。

次からIPL起動のBASICプログラム？です。

2700ボーというボーレートは、PCやFMがプログラムを読み終える前にBASICとプログラム本体を読み終える速さなのです。スゴイでしょう！

これが、UNLISTをかけたBASICをセーブする場合にも使えるのです。コンプティークの"カムイの剣"とかボーステックの"トロップウェイ"などがそうですが、これは次のような方法でできるのです。

1．BASICを起動してプログラムをロードする。

2．ブレーク禁止、UNLIST化する。

3．プログラム（テキスト）の終わりを示すポインタ（各BASICで異なる。前述の表3を参考のこと）の中身を見てプログラムの終わりを捜す。

4．次のようにして、そっくりセーブする。

KEY0, "RUN"+CHR$ (13) : SAVEM"CAS : file name, 0, &H××××

（××××はプログラムの終わりの番地）

プログラムの終わりが分からない場合、

～SAVEM "CAS : file name", 0, &HFEFF

としてしまえば、多少ロード時間が長くなりますが間違いありません。

こうしてセーブしたプログラムはIPLから一発でロードでき、オートランがかかります。もちろんUNLISTも残ったままです。

SAVEが使えない時は、モニタからセーブしてください。

＊S0000 XXXX 0000 : file name

この場合、オートランはかかりません。また、INPUT 文で受付ける最大文字数が80文字までになりますが、実用上さしつかえないでしょう。

〈表 8〉 NEWON nで削られるコマンドと増えるフリー・エリア

<table>
<tr><th>NEWONの値</th><th>CZ-8CB01 V2.0</th><th>CZ-8FB01 V2.0</th><th>CZ-8FB02 V1.0</th></tr>
<tr><td>何もしない</td><td colspan="2">20423バイト
DTL, CANVAS, LAYER, WAIT, BIN$, OCT$, CHARACTER$
SIZE, STRPTR, POKE@, PEEK@, DEVF, FAC, SUM, REPEAT-UNTIL
DEFKEY, CURSOR, GRAPH, MUSIC, TEMPO, VERIFY, CLR, LIMIT, KLIST
TEXT VRAMのLINE/BOX/BOXFILL/GET@/PUT@</td><td>75517バイト
全コマンド使用可</td></tr>
<tr><td>9</td><td colspan="2">20881バイト
TVPW, CHANNEL, VOL, CSRLIN, SPACE$, SGN, POS
KEYLIST, KEYLLIST, RANDOMIZE, MIRROR$</td><td>76216バイト
MIRROR$, KANJI$, DTL, RANDOMIZE, WAIT,
KEYLLIST, KBUF, CANVAS, LAYER, TVPW, KEYLIST, KLIST, CHANNEL, VOL</td></tr>
<tr><td>8</td><td>24162バイト
KILL, DEVI$, DEVO$, FPOS, SET, ATTR$,
RSET, LSET, GET, PUT,
OPTION SCREEN
"MEM:","0:","3:" "EMM:"
NAME, DEVF, LOF, FIELD</td><td>21476バイト
APSS, FAST, REW, EJECT, CMT, LOAD ?, "CAS:"</td><td>76872バイト
VERIFY, LOAD ?, "CAS:", CMT, CMT 関数, REW, FAST, EJECT, APSS</td></tr>
<tr><td>7</td><td>24369バイト
SCROLL, STRIG, STICK, CLICK, REPEAT ON/OFF</td><td>21790バイト
SCROLL, STRIG, STICK, ATTR$, CLICK, REPEAT ON/OFF</td><td>79405バイト
"COM:", ON COM GOSUB, COM ON,
COM OFF, COM STOP, ATTR$,
RUN"?. Sys", STICK, STRIG,
PUSH, POP
.POSITION, PATTERN, CIRCLE @, SCROLL</td></tr>
<tr><td>6</td><td>24556バイト
CRT, ON KEY GOSUB, KEY ON, KEY OFF</td><td>21977バイト
CRT, ON KEY GOSUB, KEY ON, KEY OFF</td><td>79795バイト
ON KEY GOSUB, KEYnON, KEYnOFF, KEYnSTOP
CRT, TIME$OFF, TIME$STOP : ON TIME$=""
ON TIME$="HH : MM/mm"
GOSUB, TIME$ON</td></tr>
<tr><td>5</td><td>24648バイト
APSS, FAST, REW, EJECT</td><td>22269バイト
FPOS, SET, NAME</td><td>80671バイト
MKDIR, CHDIR, RMDIR, HDOFF, SET, NAME, FPOS</td></tr>
<tr><td>4</td><td>25773バイト
HCOPY, TIME$, DAY$, DATE$, TIME, LPOS</td><td>23394バイト
HCOPY, TIME$, DATE$, TIME, LPOS</td><td>81054バイト
MOUSE, MOUSE 関数, HCOPY</td></tr>
<tr><td>3</td><td>26539バイト
VARPTR, GET@, PUT@, CGPAT$, MEM$, PAUSE, KEY0</td><td>24305バイト
VARPTR, GET@, PUT@, CGPAT$, DEVI$, DEVO$, MEM$, PAUSE, KEY0</td><td>82425バイト
GET@, PUT@, CGPAT$, DEVI$, DEVO$, COPY, LPOUT, CONSOLE #</td></tr>
<tr><td rowspan="2">2</td><td>28381バイト</td><td>26204バイト KILL</td><td rowspan="2">84934バイト
LIST, LLIST, DELETE, RENUM, AUTO, EDIT
KILL, CONT, エラーメッセージ
TRON, TROFF, SAVE, SEARCH</td></tr>
<tr><td colspan="2">TRON, TROFF, SAVE, LIST, LLIST, SEARCH, DELETE
RENUM, AUTO, EDIT, CONT, エラーメッセージ</td></tr>
<tr><td rowspan="2">1</td><td>29024バイト</td><td>26887バイト</td><td rowspan="2">85120バイト
PLAY, MUSIC, TEMPO, SOUND, CBLACK</td></tr>
<tr><td colspan="2">SOUND, PLAY</td></tr>
<tr><td rowspan="2">0</td><td>33640バイト</td><td>31503バイト</td><td rowspan="2">86883バイト
WINDOW, LINE, PSET, PRESET, CIRCLE, POLY, PAINT, SYMBOL, POINT
PRW, PALET, CLICK, NEWONn</td></tr>
<tr><td colspan="2">NEWONn, POINT, WINDOW, PSET, PRESET
LINE, PAINT, CIRCLE, POLY</td></tr>
</table>

マシン語のバックアップ・テクニック



最近のＸ１はソフトはレベルが非常に高くなっていて、一部のソフトを除くほとんどがオール・マシン語で作られています。（中にはエニックスの〝ポートピア殺人事件〟などのようにＸ１だけマシン語のものままであるります。

オールマシン語のソフトはＩＰＬ起動といって、テープをセットして電源を入れるだけでロードが始まりオート・ランをします。ミーハーな〝ロードランナー〟（LOADしてRUNするだけの人）にとっては、とてもありがたい機能ですが、我々ハッカーにとっては困りものなのです。

では、このようなソフトのバックアップを取るにはどうしたら良いのでしょうか？それを知るには、ソフト・テープに何が書かれているのか、つまり「テープ・フォーマット」のことを知らなければなりません。

2700ボーはＸ１だけ!!

Ｘ１のテープのボーレートは2700ボーで、ＦＭ（1600ボー）、ＰＣ（600、1200ボー）、ＭＳＸ（1200、2400ボー）などと比べても最高速です。

2700ボーとは１秒間に2700ビ

〈リスト17〉 ポートピアのソース・リストの一部。開発言語はいったい何なのだろう？？

```
            !("「オカケニ ナッタ ハ゛ンコ゛ウハ ケ゛ンサ゛イ"OD)
            !(" ショウサレテ イマセン。"OD00):RET
LP4130      !PRINT:!(" 「コウヘ゛ノ シカ゛イキョクハ゛ンハ 078。"OD)
            !("   キョウトハ 075 テ゛ス。"OD00):!LP1650:RET
LP4140      A=2:!COLOR:!PRINT
            !(" 「コラーッ テ゛カカ゛ 110ハ゛ン マワシテ"OD)
            !("  ト゛ナイスルンシ゛ャイッ!!(カ゛チャン)"OD00)
            !BEEP:BC=500:!WAIT:A=7:!COLOR
            !PRINT:!("ヤス 「ヤーイ オコラレタッ!!"OD00)
            !LP1650:RET
LP4160      !PRINT:!("*「ハイッ テラタ゛ヤ リョカン テ゛スカ゛・・・"OD00)
            !LP1650:A=(OD+5):IF A<>0 THEN !LP5340:GOTO LP4180
            !PRINT:!("ホ゛ス? "0000):!LP1150A:GOTO LP4060
LP4180      A=6:!COLOR:!PRINT
            !("*「アア ヒラタサン? タシカニ ウットコニ キマシタ。"OD)
            !("  ナンテ゛モ アミタ゛カ゛ネヘ イクユウテ・・・・"OD00):!LP1650
LP4190      !LP5340:!PRINT
            !("*「エッ? マタ゛ カエッテ イヤラヘンノト゛スカ・・?"OD00):!LP1650
            !PRINT:!("  サアネエ・・・・・"OD00):RET
LP4210      A=(HENSC):IF A=2 GOTO LP4220
            A=7:(HENMS)=A:!LP4800:RET
LP4220      !PRINT:!("*「ハイッ ライライケン テ゛スカ゛・・・?"OD00)
            !LP1650:BC=800:!WAIT
            !PRINT:!("ヤス 「ヤァ ヤス タ゛ケト゛。"OD00)
            !LP1650:BC=800:!WAIT
            !PRINT:!("*「コレハ マイト゛!! テ゛、コ゛チュウモンハ・・?"OD00)
            !LP1650:!PRINT:!("ホ゛ス? "0000):!LP1140A
```

ットの情報を送ることのできる速度であることを示しています。これらの情報をテープを通して正しく読み書きするには、どこからどこまでが準備信号で、どういう信号が来たらデータの始まりなのかといった手順を送りSAVE 側と受け入れる LOAD 側で申し合わせておかなくてはなりません。

これがテープ・フォーマットで、この手順が少しでも違うとデータを正しく読むことができないのです。そこでX1には標準フォーマットが決まっていて、BASICのプログラムやマシン語プログラムのテープも標準フォーマットに従って書かれています。

プロテクトがかけられたソフトというのは標準以外のフォーマットで書かれているのです。X1の1PLは標準以外のフォーマットは読めませんので、先頭に標準フォーマットで書かれたプログラム(ローダーと呼ぶ)がまず読み込まれ、このローダーによってプロテクトのかけられた本体が読み込まれるようになっています。

マシン語のプロテクトについては、後ほど詳しく見ていくことにして、ここではX1の標準フォーマットについて見ていくことにしましょう。



●X1のテープの標準フォーマット

プログラムのテープをラジカセにかけて聞いてみると、大きく分けて2つの部分からできていることが分かります。

このうち始めの部分をFCB(ファイル・コントロール・ブロック)とかインフォメーション・ブロックと呼び、後の長い部分をデータ・ブロックと呼んでいます。(図5)

FCBはファイルの種類やファイル・ネームなどの32バイトの情報を含んでいます。図6にFCBとデータ・ブロックの詳しい構造を示します。

X1の場合、BASICプログラム、マシン語、ASCデータ・ファイルとも共通のフォーマットになっていて統一がとれています。本書パート①のFM/PC/MSXのフォーマットと見比べてください。X1のが一番マトモでしょう!?

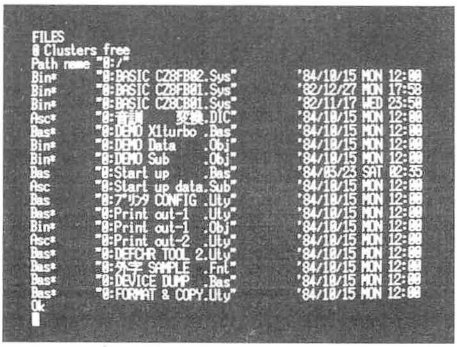
32バイトのFCBというのは、MZを除くと最長で、その結果ファイル・ネームなどは13文字+拡張3文字の計16文字まで許されていますし、パスワードを付けて秘密化することも可能です。

MS-DOSのようなタイム・スタンプ(制作年月日、時間を記録する)機能までがなんと付いています。FCBの構造はテープだけでなく、フロッピー・ディスクやハード・ディスクに至るまで統一されています。X1はテープだけを特別扱いしていませんから、ランダム・ファイルを除けば全てフロッピー・ディスクと同格に扱えます。また、2700ボーというボーレートも驚異的な速さです。

データ・ブロックの長さはFCBの19バイト目からL、Hの順で書かれています。図5を見ても分かるように、データ・ブロックの構造はBASICとマシン語では、全く同じになっていますので、後で紹介するバックアップ・ツールでも統一して扱うことができるのです。

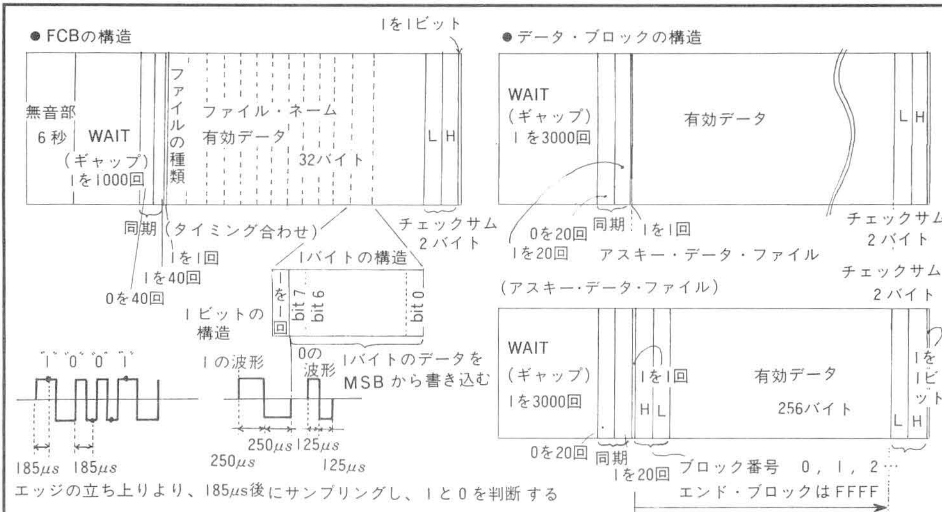
〈図5〉X1のテープ・フォーマットの大まかな構造





タイム・スタンプ付きFILESも可能。

〈図６〉



●FCBの内容

1バイト目／ファイルの種類（01H：マシン語　02H：BASICプログラム　04H：アスキー・データ・ファイル）
2～14 〃 ／ファイル・ネーム　13文字
15～17 〃 ／拡張 ファイル・ネーム　3文字
18 〃 ／パスワード
（例：BASIC CZ8CB01・V20；ワ／ → ファイルネーム　拡張子　パスワード）
19～20 〃 ／データ・ブロックの大きさ(SIZE)/L、Hの順、ASCファイルは 00 00
21～22 〃 ／ロード開始アドレス(START)/L、Hの順、マシン語以外は 00 00
23～24 〃 ／オートスタート・アドレス(AUTO)/L、Hの順　〃
25～29 〃 ／作成年月日、時分のデータ
30～32 〃 ／（ファイルの入っているトラック番号）テープの場合、00 00 00

ＡＳＣデータ・ファイルの場合、データ・ブロックが258バイトごとに区切られています。先頭の２バイトがブロック番号で、ＨＬの順で書かれており、256バイトがデータの本体です。

ブロック番号がFFFFHになっていると、データの終わりと判断されます。

●IOCS内のカセット・ルーチンを使う

IOCS内のカセット・ルーチンは全てIOCS内にまとめて作り込まれています。標準フォーマットでテープの読み書きをする場合には、IOCS内のルーチンをコールするだけで良いので簡単です。前述の表４（Ｐ.105）にこれらのルーチンのエントリー・ポイントがまとめてありますので、そちらをごらんください。

Ｘ１のBASICは、ＩＯＣＳ（0000H～0FE1H）の部分だけを切り離して単独で使えるようになっています。（漢字BASICは除く）

IOCSだけでは使いにくいので、モニタ（1000Ｈ～149ＦＨ）まで含めて分離する方法がＯｈ／ＭＺやＸ１の関連図書の中でいろいろ紹介されていますが、この中でもOh! MZ'84年７月号の"Hu-Monitor"は、ほんの少しの改造で簡単なコピー・ツールになっていて、とても便利でした。

ここでも簡単な切り離し法を紹介しておきましょう。（図７）

Ｘ１のIOCSは、とてもしっかりとした作りになっていますから、標準フォーマットで書かれたプログラムなら、IOCS内のルーチンをコールするだけで強力なコピー・ツールができあがってしまいます。標準以外のフォーマットも基本的には標準フォーマットに従っているものが多いので、IOCSを少し改造するだけで読めてしまうことがたびたびあります。

では実際にIOCSルーチンを使ってみましょう。

●FCBをのぞいてみよう

マシン語のテープのバックアップを取ったりディスクに転送しようとすると、ロード開始アドレス、マシン語の大きさ、オートスタート・アドレスの3つが分かっていないとどうしようもありません。

これらのデータはファイル・ネームとともにFCBに書かれていますから、IOCSのLOAD1ルーチンを使ってFCBの内容を表示するルーチンを組んでみましょう。

図7のMONITORを起動して、リスト18を打ち込んでください。

この例ではFCBバッファとして1480Hから32バイトを使っています。標準フォーマットではFCBの長さは0020Hバイトですので、BCレジスタに0020H、HLレジスタに1480Hをセットして LOAD 1をコールします。

FCBのロードが終わってもカセット・メカはストップしませんから、ACCに01H(STOP)をセットしてCMT COMをコールしてメカを止めています。

3つのパラメータ、SIZE、START、AUTOはそれぞれ1492H〜、1494H〜、1496H〜の2バイトにL、Hの順に入っていますので、これを表示しています。リストを打ち終えたら、テープをセットし＊G14A0 ⏎で実行してください。

皆さんのお手持のマシン語プログラムのパラメータを調べてみてください。手元にあるマシン語プログラムを調べてみると次のようになっていました。

①ベジタブルクラッシュ
SIZE、START、AUTO
3000、C800、E800
② BASIC CZ-8CB01
9E10、0000、0000
③サンダーフォース
FF00、0000、0000
④プラズマライン(ワッ!プロテクトだ!)
0040、FFC0、0000

〈図7〉モニターの切り離し方 IOCS

```
:0100=1A 3E 82 ED 79 0B ED 78 /.>わy.
:0108=3E 70 ED 79 0B ED 78 CD />p  .  ヘ
:0110=A1 09 CD 98 09 CD 5A 0A /。.ヘ※ヘZ.
:0118=CD 26 0B ED 5E 21 46 03 /ヘ&.   !F.
:0120=22 52 00 CD 2D 01 CD 3C /"R.ヘ-.ヘ<
:0128=01 FB C3 A0 14 21 52 00 /. ※  .!R.
:0130=7C ED 47 3E E4 CD FE 0D /|  >   ※
:0138=7D C3 54 0B 01 00 1C 3E /}テT....>
:0140=07 ED 79 05 3E 3F ED 79 /.   .>?
```

(A0 14 →) 0010にする

```
:1038=CD EC 0D 18 C3 5E 23 56 /ヘ※.テ #V
:1040=D5 D9 C9 44 8B 11 4D 1D /ユル/D※M.
:1048=12 50 61 10 47 86 11 46 /.Pa.G※F
:1050=53 12 52 F0 10 53 6A 10 /S.R※Sj.
:1058=4C 9A 10 56 E1 10 54 BE /L※V※Tセ
```

(F0 10 →) 0010にする

CZ-8(F)CB01を起動する
上記のようになおして、
＊S0000 147F 0000 : Hu-Monitor ⏎ とする。Hu-Monitorは1480H〜149FHとFF00H〜FFFFH まではワーク・エリアになっている
それ以外は全てフリー・エリアとして使える

〈リスト18〉マシン語のアドレス表示プログラム(FCBをのぞいてみよう)

```
:14A0=21 80 14 01 20 00 CD 41 /!※.  .ヘA
:14A8=00 11 62 14 CD 21 13 3E /..b.ヘ!.>
:14B0=01 CD EC 0D CD DD 14 2A /.ヘ※ヘン.*
:14B8=92 14 CD 02 12 CD DD 14 /※ヘ..ヘン.
:14C0=2A 94 14 CD 02 12 CD DD /*※ヘ..ヘン
:14C8=14 2A 96 14 CD 02 12 3E /.*※ヘ..>
:14D0=0D CD 13 00 11 80 14 21 /.ヘ...※!
:14D8=9F 14 C3 97 11 3E 2C C3 /※テ※>,テ
:14E0=13 00 00 00 00 00 00 00 /........
:14E8=00 00 00 00 00 00 00 00 /........
```

〈リスト19〉

IOCS マシン語バックアップツール　Loader Saver/ソース・リスト

```
0000              ; LOADER SAVER V.1
0000              ; LOAD:14A0H   SAVE:14A3H
0000              ; OBJECT    1500H--
0000              ;
14A0                      ORG  14A0H
14A0              ;
14A0 C3 BE 14     LOAD:   JP   LOADL
14A3 21 80 14     SAVE:   LD   HL,1480H        ← ❶
14A6 01 20 00             LD   BC,0020H        ← ❷
14A9 11 5A 14             LD   DE,145AH        ← ❸
14AC CD 21 13             CALL 1321H
14AF CD 3B 00             CALL 003BH           ← ❹
14B2 21 00 15             LD   HL,1500H
14B5 ED 4B 92             LD   BC,(1492H)      ← ❺
14B8 14
14B9 D4 3E 00             CALL NC,003EH        ← ❻
14BC 18 19                JR   CSTOP
14BE              ;
14BE 21 80 14     LOADL:  LD   HL,1480H
14C1 01 20 00             LD   BC,0020H
14C4 CD 41 00             CALL 0041H           ← ❼
14C7 11 62 14             LD   DE,1462H        ← ❽
14CA CD 21 13             CALL 1321H
14CD 21 00 15             LD   HL,1500H        ← ❾
14D0 ED 4B 92             LD   BC,(1492H)
14D3 14
14D4 D4 44 00             CALL NC,0044H        ← ❿
14D7              ;
14D7 11 73 14     CSTOP:  LD   DE,1473H        ← ⓫
14DA DC 0B 00             CALL C,000BH         ← ⓬
14DD 3E 01                LD   A,01            ← ⓭
14DF C3 EC 0D             JP   0DECH           ← ⓮
14E2              ;
14E2                      END
```

❶FCBバッファを1480Hとする
❷FCBの長さは0020H
❸Writingのメッセージ・アドレス
❹FCBのセーブ
❺データ・ブロックの長さ
❻データ・ブロックのセーブ
❼FCBのロード
❽Foundのメッセージ・アドレス
❾データ・ブロックを置くアドレス
❿データ・ブロックのロード
⓫ERR？のメッセージ・アドレス
⓬文字列表示
⓭ストップ
⓮カセット・コントロール

●マシン語のバックアップ・ツール

今度は LOAD 2 ルーチンまで使ってマシン語の本体までロードさせてみましょう。

マシン語本体の長さは、ＦＣＢの19〜20バイト目に書かれています。ＦＣＢバッファのアドレスを1480Ｈにしましたので、1492Ｈから２バイトが SIZE になります。LOAD 2 ルーチンの使い方は LOAD 1 とほとんど同じで、ＨＬレジスタにロード開始アドレス、ＢＣレジスタに SIZE を入れてコールします。

ロード開始アドレスはＦＣＢの内容をそのまま使うのではなく、1500Ｈにします。そうすると、どの番地からロードするプログラムでも1500Ｈ以降に置けるからです。リスト20を打ち込んだら、

＊ G14A0 ↵

でロードが始まり、

＊ G14A3 ↵

でセーブされます。

ＦＣＢの詳しい内容を知りたい時は、

＊ D1480　149F ↵

でＦＣＢがのぞけます。

1500Ｈからダンプするとデータの本体を見ることができます。必殺人間逆アセンブラと呼ばれる人種は、これだけでマシン語の解析ができるそうです。

なお、このプログラムはマシン語だけでなく BASIC プログラムのバックアップもできます。（ＡＳＣデータ・ファイルはできません）

ＮＥＷしたプログラムを復活する方法のところでＸ１ターボの漢字BASICでとりあえずマシン語としてセーブしてあったBASICプログラムは、

＊ G14A0

でロードした後、ＦＣＢバッファの先頭（1480Ｈ＝01H）の01H を 02H (BASIC) に書き換え、

＊ M1480 ↵

：1480＝01⇨02　SHIFT＋BREAK

＊ G14A3 ↵

でセーブすれば、BASIC プログラムを復活することができます。

リスト20 のバックアップ・ツールのフリー・エリアは1500Ｈから FEFFＨまでとなっています。（FF00H 以降はキー入力バッファやシステム・スタック・ポインタなどのワーク・エリアです）

従って、FF00H −1500H＝EA00H までの長さのマシン語プログラムまではバックアップできますが、これ以上長いプログラム（例えばサンダーフォース）になるとメイン・メモリと G-RAM とに分けてロードするというハイテクニックが必要となりますので、このツールではバックアップできません。

なお、このバックアップ・ツールをＩＰＬ起動で使う場合。

＊ M012B ↵

：012B＝0010 ↵

とした後、SHIFT＋BREAK し、

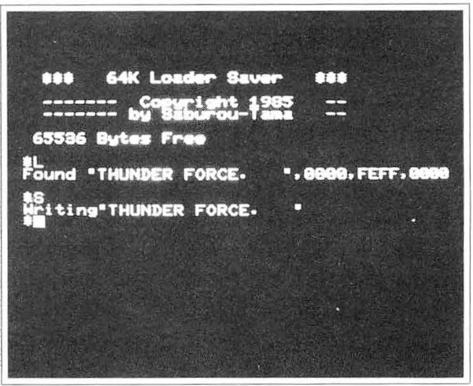
＊ S0000　14FF　0000：ファイル・ネーム ↵

でセーブすれば、次からはＩＰＬ起動で使えます。

〈リスト20〉マシン語のバックアップ・ツール

```
:14A0=C3 BE 14 21 80 14 01 20 /テセ.!※.
:14A8=00 11 5A 14 CD 21 13 CD /..Z.^!.^
:14B0=3B 00 21 00 15 ED 4B 92 /;.!..
:14B8=14 D4 3E 00 18 19 21 80 /.ヤ>...!
:14C0=14 01 20 00 CD 41 00 11 /.. .^A..
:14C8=62 14 CD 21 13 21 00 15 /b.^!.!..
:14D0=ED 4B 92 14 D4 44 00 11 /   ※ヤD..
:14D8=73 14 DC 0B 00 3E 01 C3 /s.ワ..>.テ
:14E0=EC 0D 00 00 00 00 00 00 /※......
:14E8=00 00 00 00 00 00 00 00 /........
```

最強のコピーツール TAPE⇨TAPE "64K Loader Saver"



X1のマシン語ソフトは、バックアップが難しいこともあって、特にこれといったプロテクトがかかっていないソフトも結構あります。最近のソフトのように40Kバイト以上のものになると、ただデカイ、長いだけで一種のプロテクトにもなるからです。

そこで先ほどのようなコピー・ツールが使われるわけですが、以前はEA00Hというフリー・エリアは必要にして十分な大きさでした。しかし、時代はついに0000HからFEFFHまで一発でロードするというプログラムを登場させたのです。〝サンダーフォース〟がそうです。プロテクトらしいプロテクトをかけずに、我々に手も足も出させないというものです。このプログラムはハッカーに対する挑戦でした。

しかし、これに負けているわけにはいきません。X1には48KバイトものG-RAMがあるのです。メイン・メモリと合わせれば100Kバイト以上のフリー・エリアになるのです。

そこで〝メイン・メモリとG-RAMに分けロードする〟という必殺のテクニックが登場するのです。

いくつかの試行錯誤が繰り返えされましたが、いろいろ煮つめていくうちに、最もセコイ〝IOCSルーチンにパッチを当てる〟という方法で解決することになりました。（図7参照のこと）

また、コンパクトなプログラムですが、ハドソン・シリーズのプロテクト・ファイラーが付いていますので、ハドソン・シリーズならロードしてセーブするだけでプロテクトが、はずれます。

●X1"Loader Saver"の実体は?

この〝64K Loader Saver〟ではメイン・メモリの2000H〜5FFFHまでと、G-RAMの4000H〜FFFFHまでを連続した64Kバイトの仮想メモリに対してロード、セーブが行なわれます。

また、BASICのIOCSのモニタ部を使っていますので、モニタの持つコマンドは全て使えます。D（ダンプ）コマンドやその他のコマンドも全てこの仮想メモリに対して行なわれます。（図8参照）

例えば、

＊D0000 ↵

とすると、見た目は0000Hからダンプしているようですが、実際には2000H〜の内容をのぞいているわけです。使う側は仮想メモリであることをまったく意識しないですむようになっているので、使ってる内に0000H〜FFFFHまでの全てがフリー・エリアのような錯覚を起してしまうでしょう。

表9に〝64K Loader Saver〟の全てのコマンドと使い方をまとめておきます。

プログラムを入力するにはまず、旧BASICかNEW-BASICを起動してチェックサム付きダンプ・プログラムをRUNさせて、リスト21を打ちこんでください。

次にMON ↵でモニタに移り

＊TC000 C3FF 14A0 ↵

＊G 14A0 ↵

とすると、モニタが改造され〝64K Loader Saver〟が起動します。このプログラムが仮想メモリの2000H以降に転送されていますので、

＊S2000 37DF 0000 0000：64K L & Saver ↵

とすれば、〝64K Loader Saver〟がセーブされ、次からはIPL起動で使うことができます。

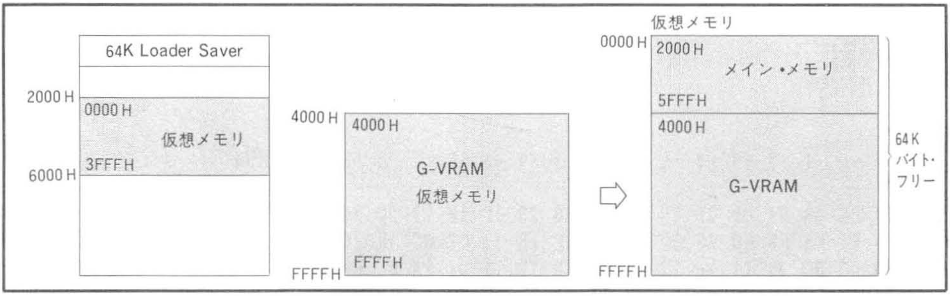
〈図7〉 64Kバイトを1発セーブにするパッチ

```
0BBD                  ;
0BBD C5          WLOOP:   PUSH BC
0BBE E5                   PUSH HL
0BBF CD D6 0C             CALL SUMCL
0BC2 7E          WLP1:    LD   A,(HL)
0BC3 CD 8A 0C             CALL W1BYTE
0BC6 38 17                JR   C,BREAK?
0BC8 23          WLP2:    INC  HL
0BC9 0B                   DEC  BC
0BCA 78                   LD   A,B
0BCB B1                   OR   C
0BCC 20 F4                JR   NZ,WLP1
0BCE 2A A1 0E             LD   HL,(CSUM)
0BD1 7C                   LD   A,H
0BD2 CD 8A 0C             CALL W1BYTE
0BD5 7D                   LD   A,L
0BD6 CD 8A 0C             CALL W1BYTE
0BD9 CD A5 0D             CALL L1CYCL
0BDC AF                   XOR   A
0BDD 18 14                JR    RETOK
0BDF                  ;
0BDF 3E E6       BREAK?:  LD    A,0E6H
0BE1                  ;
```

パッチ → CALL XSAVE

```
XSAVE:   CALL LDAHL
         JP   W1BYTE
;
;
LDAHL:   PUSH HL
         EXX
         POP  BC
         LD   A,B
         CP   40H
         JP   C,MEMRD
         IN   A,(C)
         EXX
         RET
;
MEMRD:   ADD  A,20H
         LD   B,A
         LD   A,(BC)
         EXX
         RET
```

〈図8〉 64K Loader Saver のメモリ・マップ



〈表9〉 "64K Loader Saver" のコマンド

- D、M、F、P、T、L、Vコマンドとも Hu－BASIC のモニタと使い方は同じですが、全て仮想メモリに対して行なわれます。
- G〔add〕
  仮想メモリをメインメモリに移して、指定したアドレスにジャンプします。RET はできません。
- S
  単にS⏎とすると、直前にL⏎で読み込んだそのままをセーブします。
- S〔ad 1〕〔ad 2〕〔ad 3〕: filename
  BASIC のモニターの S コマンドと使い方は同じです。ad 1 から ad 2 まででオートスタートは ad 3 としてセーブします。
- S〔ad 1〕〔ad 2〕〔ad 3〕〔ad 4〕: filename
  ad 1 から ad 2 までをオートスタート ad3、ロード開始アドレス ad4 でセーブします。
- ＃ WIDTH40と80を切りかえる
- ＃ C グラフィックを消す
- ＃ Pグラフィックを表示する

〈リスト21〉 "64K Loader Saver" プログラム

```
ｱﾄﾞﾚｽ +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  CSUM  [  ASCII CODE  ]
--------------------------------------------------------------------------------
:C000=C3 E9 17 CD 15 17 23 C3 14 11 CD 2A 17 CD 07 12  :BB   ﾃ..ﾍ..#ﾃ..ﾍ*.ﾍ..
:C010=C9 CD 2A 17 FE 20 C9 1A B7 20 21 3A 07 00 FE 50  :5F   ﾉﾍ*.. ﾉ.ｷ !:...P
:C020=30 0D 3E FF 32 9D 11 3E 10 32 A2 11 C3 8C 09 3E  :23   0.>.2..>.2｢.ﾃ..>
:C030=7F 32 9D 11 3E 08 32 A2 11 C3 98 09 CD 50 14 FE  :1D    2..>.2｢.ﾃ..ﾍP..
:C040=20 28 F9 0E 10 FE 43 28 12 FE 50 C0 3E AA ED 79  :36    (....C(..Pﾀ>ｪ.y
:C050=04 3E CC ED 79 04 3E F0 ED 79 C9 AF ED 79 04 ED  :DB   .>ﾌ.y.>..yﾉｯ.y..
:C060=79 04 ED 79 C9 CD 0C 15 DA 78 14 C9 F5 2A 94 14  :90   y..yﾉﾍ..ﾚx.ﾉ.*..
:C070=CD 24 15 ED 5B 92 14 19 2B CD 24 15 2A 96 14 CD  :DF   ﾍ$..[...+ﾍ$.*..ﾍ
:C080=24 15 F1 C9 3E 2C CD 13 00 C3 02 12 D9 E5 C5 D9  :70   $..ﾉ>,ﾍ..ﾃ..ﾙ.ﾅﾙ
:C090=CD 41 00 D9 C1 E1 D9 F5 C5 E5 01 20 FF 0D 0A ED  :25   ﾍA.ﾙﾁ.ﾙ.ﾅ.. ....
:C0A0=79 20 FA 60 69 11 80 14 01 20 00 ED B0 E1 C1 F1  :52   y . i.... ..ｰ.ﾁ.
:C0B0=C9 7C 3C 20 28 7D B7 20 46 21 00 60 CD 44 00 D8  :CD   ﾉ|< (}ｷ F!. ﾍD.ﾘ
:C0C0=01 12 FF 2A 12 60 22 92 14 E5 CD 31 16 2A 14 60  :0D   ...*. "...ﾍ1.*.
:C0D0=22 94 14 22 96 14 CD 31 16 CD 31 16 C1 3E CD 32  :BC   ".."..ﾍ1.ﾍ1.ﾁ>ﾍ2
:C0E0=15 0C E5 21 99 15 22 16 0C E1 CD 44 00 21 15 0C  :4D   ...!.."...ﾍD.!..
:C0F0=36 77 23 36 23 23 36 0B C9 CD 15 17 23 0B C9 FE  :44   6w#6##6.ﾉﾍ..#.ﾉ.
--------------------------------------------------------------------------------
CSUM  46 9E 25 1A 24 84 F4 23 FB 2B 5C EC 47 37 0A 10   E8
```

```
ｱﾄﾞﾚｽ +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  CSUM  [  ASCII CODE  ]
--------------------------------------------------------------------------------
:C100=DE 20 DA CD 7D 15 D8 21 08 20 7D AE 77 2C 20 FA  :40   ﾞ ﾚﾍ}.ﾘ!. }ｮw, .
:C110=01 FA FF ED 78 32 26 0D 3E 1B 32 C7 0C 3E 20 32  :B2   ....x2&.>.2ﾇ.> 2
:C120=C2 0C 21 11 16 22 0A 0C CD 7D 15 D4 AA 0C 6F D4  :7A   ﾂ.!..."..ﾍ}.ﾔｪ.oﾔ
:C130=AA 0C 67 22 96 14 3E 14 32 26 0D 3E 13 32 C7 0C  :F6   ｪ.g"..>.2&.>.2ﾇ.
:C140=3E 28 32 C2 0C 21 D6 0C 22 0A 0C D8 21 92 14 01  :41   >(2ﾂ.!ﾖ."..ﾘ!...
:C150=12 FF 16 06 7E ED 79 23 03 15 20 F8 2A 94 14 ED  :23   ....‾.y#.. .*...
:C160=4B 92 14 CD 2A 17 AD CD 15 17 23 0B 78 B1 20 F3  :0F   K..ﾍ*.ｭﾍ..#.xｱ .
:C170=C9 D1 E1 C1 CD AA 0C 4F CD AA 0C 47 CD AA 0C 6F  :CA   ﾉﾑ.ﾁﾍｪ.Oﾍｪ.Gﾍｪ.o
:C180=CD AA 0C 67 22 94 14 ED 43 92 14 C5 E5 D5 C3 D6  :A2   ﾍｪ.g"...C..ﾅ.ﾕﾃﾖ
:C190=0C 7D ED 79 03 7C ED 79 03 C9 CD A6 12 D0 1A B7  :C6   .}.y.|.y.ﾉﾍｦ.ﾐ.ｷ
:C1A0=37 C0 E1 21 A0 14 01 20 FF 2B 0D F5 ED 78 77 F1  :C7   7ﾀ.! .. .+...xw.
:C1B0=20 F7 11 5A 14 CD 21 13 01 20 00 CD 3B 00 DA 78  :12    ..Z.ﾍ!.. .ﾍ;.ﾚx
:C1C0=14 2A 94 14 ED 4B 92 14 CD 8B 16 DA 78 14 3E 01  :D7   .*...K..ﾍ..ﾚx.>.
:C1D0=CD EC 0D C9 C5 E5 21 A0 14 01 20 FF 2B 0D F5 7E  :D9   ﾍ..ﾉﾅ.! .. .+..‾
:C1E0=ED 79 F1 20 F7 E1 C1 CD 3B 00 C9 E5 21 9D 16 22  :BC   .y. ..ﾁﾍ;.ﾉ.!.."
:C1F0=C4 0B E1 CD 3E 00 21 8A 0C 22 C4 0B C9 CD 2A 17  :3A   ﾄ..ﾍ>.!.."ﾄ.ﾉﾍ*.
--------------------------------------------------------------------------------
CSUM  71 34 FC 68 E2 4E 06 3D BA 12 DD 9F 7C D1 6B 0A   86
```

```
ｱﾄﾞﾚｽ +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  CSUM  [  ASCII CODE  ]
-------------------------------------------------------------------------------
:C200=C3 8A 0C CD AA 0C DA DF 0B 57 CD 2A 17 BA C2 31  :B2   ﾃ..ﾍｴ.ﾚﾟ.Wﾍ*.ｺﾂ1
:C210=0C C3 5E 0C CD 2A 17 EB BE EB C0 C5 D5 E5 41 CD  :28   .ﾃ^.ﾍ*..ｾ.ﾀﾅﾕ.Aﾍ
:C220=2A 17 EB BE EB 20 05 13 23 10 F4 AF E1 D1 C1 C9  :1F   *..ｾ. ..#..ｯ.ﾑﾁﾉ
:C230=CD 5E 11 3F D0 77 23 18 F7 CD 1F 11 D8 E5 DD E1  :6C   ﾍ^.?ﾐw#..ﾍ..ﾘ.ﾝ.
:C240=F3 2E 00 CD 33 01 01 20 FF ED 78 02 0C 20 FA 21  :F0   ...ﾍ3.. ..x.. .!
:C250=06 17 11 00 FF 01 0F 00 ED B0 21 00 20 11 00 00  :2C   .........ｰ!. ...
:C260=01 00 40 C3 00 FF ED B0 4B 42 ED 78 02 03 78 FE  :0D   ..@ﾃ...ｰKB.x..x.
:C270=FF 20 F7 DD E9 E5 D9 C1 F5 78 FE 40 38 05 F1 ED  :21   . .ﾝ..ﾙﾁ.x.@8...
:C280=79 D9 C9 C6 20 47 F1 02 D9 C9 E5 D9 C1 78 FE 40  :12   yﾙﾉﾆ G..ﾙﾉ.ﾙﾁx.@
:C290=38 04 ED 78 D9 C9 C6 20 47 0A D9 C9 CD 2A 17 EB  :15   8..xﾙﾉﾆ G.ﾙﾉﾍ*..
:C2A0=CD 15 17 EB 23 13 0B 78 B1 20 F1 C9 CD 2A 17 EB  :21   ﾍ...#..xｱ .ﾉﾍ*..
:C2B0=CD 15 17 EB 2B 1B 0B 78 B1 20 F1 C9 06 10 CD FB  :16   ﾍ...+..xｱ .ﾉ..ﾍ.
:C2C0=14 21 00 10 22 2B 01 E5 11 6E 17 C3 0B 00 20 2A  :26   .!.."+...n.ﾃ.. *
:C2D0=2A 2A 20 20 20 36 34 4B 20 4C 6F 61 64 65 72 20  :00   **   64K Loader
:C2E0=53 61 76 65 72 20 20 20 2A 2A 2A 0D 20 2D 2D 2D  :93   Saver   ***. ---
:C2F0=2D 2D 2D 20 20 20 43 6F 70 79 72 69 67 68 74 20  :C0   ---   Copyright
-------------------------------------------------------------------------------
CSUM  C8 07 55 0C 68 92 54 57 5C E6 E6 37 62 64 30 5C   86
```

```
ｱﾄﾞﾚｽ +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  CSUM  [  ASCII CODE  ]
-------------------------------------------------------------------------------
:C300=31 39 38 35 20 2E 39 20 2D 0D 20 2D 2D 2D 2D 2D  :B9   1985 .9 -. -----
:C310=2D 20 20 20 62 79 20 20 53 61 62 75 72 6F 75 2D  :B6   -   by  Saburou-
:C320=54 61 6D 61 20 20 2D 0D 0D 20 36 35 35 33 36 20  :53   Tama  -.. 65536
:C330=42 79 74 65 73 20 46 72 65 65 0D 0D 00 CD 1F 11  :C0   Bytes Free...ﾍ..
:C340=38 03 22 94 14 CD 8B 13 C9 3E 02 32 8B 0A 3E 18  :96   8."..ﾍ..ﾉ>.2..>.
:C350=32 66 13 3E C3 32 19 11 32 C9 12 32 D2 12 32 55  :B2   2f.>ﾃ2..2ﾉ.2ﾒ.2U
:C360=0C 21 A3 14 22 1A 11 21 AA 14 22 BC 11 22 36 12  :69   .!｣."..!ｪ."ｼ."6.
:C370=AF 32 1C 11 32 68 14 3E CD 32 CD 11 32 B0 10 21  :EA   ｯ2..2h.>ﾍ2ﾍ.2ｰ.!
:C380=B1 14 22 CE 11 21 2C 15 22 C4 10 21 51 15 22 AE  :75   ｱ."ﾎ.!,."ﾄ.!Q."ｮ
:C390=10 21 05 15 22 B1 10 21 74 16 22 8E 10 21 8B 16  :5B   .!.."ｱ.!t."..!..
:C3A0=22 93 10 21 3A 16 22 6B 10 21 A3 16 22 56 0C 3E  :6F   "..!:."k.!｣."V.>
:C3B0=23 32 52 10 21 B7 14 22 53 10 21 5C 17 22 2B 01  :0A   #2R.!ｷ."S.!¥."+.
:C3C0=21 F0 10 22 7D 10 21 DD 17 11 F0 10 01 0C 00 ED  :F0   !.."}.!ﾝ........
:C3D0=B0 21 3C 17 22 CA 12 21 4C 17 22 D3 12 21 B4 16  :98   ｰ!<."ﾊ.!L."ﾓ.!ｴ.
:C3E0=22 74 12 21 D0 16 22 63 12 21 D9 16 22 4D 10 21  :F6   "t.!ﾐ."c.!ﾙ."M.!
:C3F0=00 00 11 00 40 01 DD 17 ED B0 C7 00 00 00 00 00  :AA   ....@.ﾝ..ｰﾇ.....
-------------------------------------------------------------------------------
CSUM  12 6E 25 80 7D F8 39 7D BF 44 70 2F 43 B2 55 52   8E
```

《極秘》これを知ればX1ハッカー
市販ソフトのプロテクト



市販ソフトの中にはプロテクトがかかっているものも多くあります。

プロテクト外しという遊びは究極の遊びと呼ばれるほど高度な知識と洞察力を必要とします。いろいろなところに仕掛けられているワナをさけながら推理を働かせて目的の宝(プログラム)に到達するところなどは、まさにアドベンチャー・ゲームそのものなのです。

このアドベンチャー・ゲームは皆さんをとことんのめり込ませる不思議な魅力をもっています。あまりのめり込みすぎて、〝The BASIC〟誌の筆者みたいにならないように!!注意、注意!!してください。

●プロテクトの基本

IPLから読み込ませるソフトは必ず標準フォーマットになっていなければなりません。

そこで、プロテクトがかけられたソフトでは、一番はじめは標準フォーマットで書かれたローダーが読み込まれこのローダーで2段目以降をロードするという2段ロードの形をとっています。2段目にロードしたローダーから、さらに3段目をロード等の3段構え4段構えになっているソフトもあります。

いずれにしても、一番はじめのブロックは標準フォーマットになっていますので、1段目を解析すれば、2段目以降のロードの仕方も分かるわけです。

そこで、一般的に行なわれているプロテクトの方法について調べてみると、大きく次の3種類に分類できます。

① 1段目でオリジナルのローダーを読み込ませる。2段目以降は、このローダーを用いてロードする。(ゼビウス、フラッピー、ヴォルガード、ニュートロン、ジェルダII)

② IPLのワーク・エリア(FF00H以降)のうち、FCBバッファ(FF00H～FF1FH)に重ねてロードする。IPLがオートスタート・アドレスを読み違えることを利用する。(めだかの兄弟、ナッツアンドミルク、ポーラスターII、ペンタンの野良仕事)

③ ワーク・エリアのうち、スタック・エリア(FFFFHからアドレスの低い方に向かって積まれている)にプログラムをロードする。サブルーチンがリターン・アドレスを間違えることを利用。(サラダの国のトマト姫、デゼニランド、野球狂、トランプ狂、システムソフトウエア・コンバーター、ドアドア)

これらを詳しくみていくことにしましょう。

まず、1のタイプはオリジナルのローダーのロード開始アドレスと、そしてオートスタート・アドレスはFCBを読むとマジメ?に書かれていますので、時間と根性さえあればローダーを解析できます。

そして、オートランしないように改造した後、このローダーを使って2段目以降をロードすれば良いのです。2段ロードの目的はメモリを有効に使うためで、1段目でPCGをセットして2段目でメイン・ルーチンを読み込んでいるソフトもあり、プロテクトばかりが目的ではないことがあります。

このような場合、オリジナルのローダーもX1の標準フォーマットになっていて、コピー・

ツールで簡単にコピーできてしまう場合があります。

また、ローダーによってはメイン・メモリのほとんどいっぱいまでロードしてしまうソフトがあります。(ニュートロンなど)このような際は、モニタを置く場所がありませんから、ロードが終わったら、

①メイン・メモリの4000H〜FFFFHまでをG-RAMに転送する。

② 0000H〜3FFFHまでのプログラムを2000H〜にブロック転送する。

③IPLにジャンプし、64K Loader Saverを起動する。

とすると、中身をのぞくことができます。

次の2および3のタイプのプロテクトは主にオートスタート・アドレスをかくすために作られたプロテクトです。

まず、2のタイプのテープをお持ちでしたらFCBをのぞくプログラムでオートスタート・アドレスを調べてください。その番地から解析を始めると訳がわかりませんね。実はオートスタートはダミーなのです。

では本当のオートスタートはどこか!という前にIPLの働きを考えてみることにしましょう。

BOOTするとまず、

①FCBバッファ(FF00H〜FF1FH)にFCBがロードされる。

② FF12H (SIZE)、FF14H (START)の内容に従って本体をロードする。

③ FF16H (AUTO) に書かれたアドレスにジャンプする。

詳しくいうと、IPL ROMからFF78Hに、

FF78=01001E
(LD BC、1E00H)
FF7B=ED79
(OUT (C)、A)
FF7D=E9
(JP (HL))

が転送されて、HLレジスタにFF16Hの内容がセットされFF78Hにジャンプする。

どうですか?ピンときませんか。ここでピンとくればハッカーの素質ありです。

つまり、IPLがFF16H (AUTO)を読むのは本体がロードされた後ですから、IPLは書き換えられた後のオートスタート・アドレスを読んで実行してしまうのです。

これで本当のオートスタートが分かりますね。図9の例ではFF70Hですので、ここから解析していけば良いのです。これから先は、あなたの力でガンバッテください。

さて、3のタイプのプロテクトは最も高度な種類に属します。このプロテクトを理解するには少なくともZ-80のマシン語とIPLの中身について精通していなければなりません。

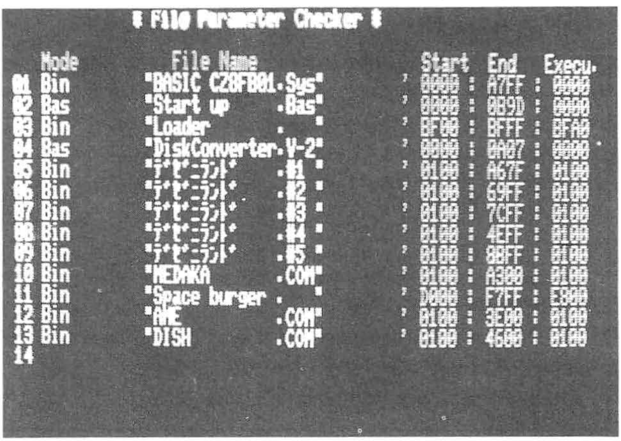
IPLの解析については、MIAのX1リファレンスノウトやOh! MZ'85年9月号を参考にすると良いと思います。

X1のIPLではスタック・ポインタは0000Hに設定されますので、スタックはFFFFHからアドレスの若い方にむかって生長していきます。

スタックには退避したレジスタの内容やサブルーチンのリターン先アドレスなどが積まれていますが、この部分にデータをロードすると暴走してしまいます。

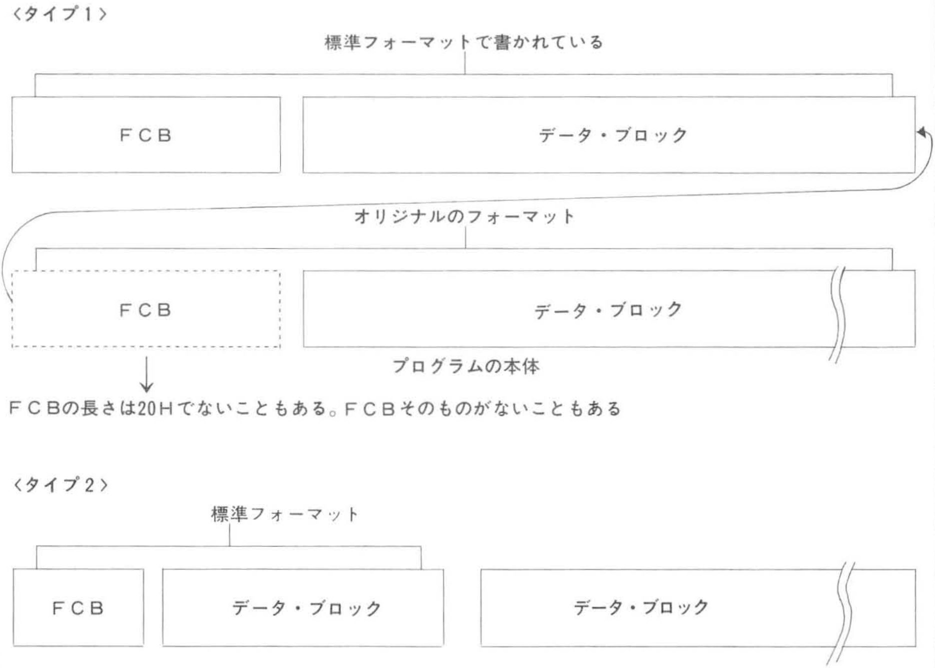
どのように暴走するかはIPLの詳しい解析が必要であるなど、いろいろゴチャゴチャありますので、結論から申しますとFFF8Hの値がHLレジスタ、FFFAHの値がBCレジスタにセットされ、FFFCHに書かれているアドレスからオートスタートします。

この時、SP(スタック・ポインタ)の値はFFFEHで、またIPL ROMはアクティブになったままです。図6の例ではFFC0H



* File Parameter Checker *

	Mode	File Name	Start	End	Execu.
01	Bin	"BASIC CZ8FB01.Sys"	0000	A7FF	0000
02	Bas	"Start up .Bas"	0000	0B9D	0000
03	Bin	"Loader . "	BF00	BFFF	BFA0
04	Bas	"DiskConverter.V-2"	0000	0A87	0000
05	Bin	"デゼニランド .#1 "	0100	A67F	0100
06	Bin	"デゼニランド .#2 "	0100	69FF	0100
07	Bin	"デゼニランド .#3 "	0100	7CFF	0100
08	Bin	"デゼニランド .#4 "	0100	4EFF	0100
09	Bin	"デゼニランド .#5 "	0100	88FF	0100
10	Bin	"MEDAKA .COM"	0100	A300	0100
11	Bin	"Space burger . "	D000	F7FF	E800
12	Bin	"AME .COM"	0100	3E00	0100
13	Bin	"DISH .COM"	0100	4600	0100
14					



このオートスタートはダミー

```
:FF00=01 B7 AC C9 DD 20 B6 DE 2D D9 20 20 20 20 20 20  :84  .ｷｬﾉﾝ ｶﾞｰﾙ
:FF10=20 20 80 00 00 FF 51 FF 82 33 23 16 42 00 00 00  :3F      ....Q..3#.B...
:FF20=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00  ................
:FF30=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00  ................
:FF40=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00  ................
:FF50=00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F1  ................
```

この部分がそっくり読み込まれる

本当のオートスタート

```
:FF00=4F 6A 69 73 61 6E 20 73 6F 66 74 20 69 6E 63 2E  :C8  Ojisan soft inc.
:FF10=00 20 7D 20 03 01 70 FF 19 7E EE 23 77 23 10 F9  :7B  . } ..p..¯.#w#..
:FF20=09 DD DC 1D 3E F0 23 08 75 08 7D 08 08 08 08 65  :B7  .ﾝﾜ.>.#.u.}....e
:FF30=08 6D CE 60 61 DC CE 70 64 DC 09 37 DC CE 4B 12  :A5  .mﾎ aﾜﾎpdﾜ.7ﾜﾎK.
:FF40=FF CD 03 00 3E 04 CD 00 00 3E 1E D3 00 2A 14 FF  :4A  .ﾍ..>.ﾍ..>.ﾓ.*..
:FF50=E9 21 00 FF 01 00 30 7E B7 CA 63 FF ED 79 23 03  :27  .!....0¯ｷﾊc..y#.
:FF60=C3 57 FF 21 00 00 11 01 00 01 00 00 36 00 ED B0  :20  ﾃW.!........6..-
:FF70=21 08 FF 11 08 00 19 19 01 00 1E ED 79 E9 00 00  :A3  !...........y...
:FF80=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0  ................
```

ここから解析をはじめる

〈図9〉

から解析を始めれば良いと思います。

以上、市販プロテクトのポイントをいくつか見てきましたが、これで市販ソフトのプロテクトが全て解けると思ったら大きな間違いです。プロテクトの解析は一筋なわではいきません。場合によっては、2重3重のプロテクトでガッチリ固めてあったりもします。

プロテクトは、これからますます発展し、難しくなっていきます。さあ、皆さんもプロテクト・アドベンチャーの世界を探険してみようではありませんか！ 皆さんのご健闘を祈ります。



〈タイプ3〉

FCB	データ・ブロック	データ・ブロック

ダミー（FF16-FF17の 00 00）

```
:FF00=01 B7 AE B3 DA C2 20 CC DF DB C3 B8 C4 20 20 20  :FA  .ｷｮｳﾚﾂ ﾌﾟﾛﾃｸﾄ
:FF10=20 20 40 00 C0 FF 00 00 FF FF FF FF FF FF FF FF  :37    @.ﾀ..........
:FF20=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00  ................
:FF30=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00  ................
:FF40=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :00  ................
:FF50=00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F1  ................
:FF60=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0  ................
:FF70=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0  ................
:FF80=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0  ................
:FF90=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0  ................
:FFA0=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0  ................
:FFB0=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0  ................
:FFC0=F3 3A 07 FF FE A0 C0 16 08 21 00 00 01 00 FF CD  :9D  .:... ﾀ..!.....ﾍ
:FFD0=F9 05 D4 F9 05 D4 85 05 D8 CD CB 05 D8 FE 50 C0  :89  ..ﾔ..ﾔ..ﾘﾍﾋ.ﾘ.Pﾀ
:FFE0=3E 04 CD 48 06 3E 1E D3 00 C7 00 00 00 00 00 00  :53  >.ﾍH.>.ﾓ.ﾇ......
:FFF0=00 00 00 00 00 00 00 00 08 00 FA FF C0 FF 00 00  :C0  ............ﾀ..
```

- ここから解析をはじめる（FFC0の F3）
- IPL ROM内のルーチン（F9 05, F9 05, 85 05, CB 05, 48 06）
- HLに入る（08 00）
- BCに入る（FA FF）
- IPL Activeのままこのアドレスにジャンプ（C0 FF）

画面で走っているソフトが1発でテープへ!! "スーパーウェポン" 64K Loader Saver II



テープ・ソフトのバックアップ総集編として皆さんに究極のバックアップ・ツール"64K Loader Saver II"を紹介します。

64K Loader Saver IIはIPLスイッチ（X1Fやターボにある赤いスイッチ）を押してIPLに戻っても、メモリの内容が消されずに残っている性質を利用したもので、メモリの内容を64K Loader Saverの仮想メモリに転送した後、前述の64K Loader Saverを起動してメモリの内容をそっくりテープに落としてしまおうというものです。

ちょうど本誌のパート①で紹介されたPC-8801用のファイルナルウエポンV3のX1版のようなものです。64K Loader Saver IIではこれを追加部品なしでやってしまおうというこです。

64K Loader Saver IIの威力をフルに発揮するには今のところIPL・スイッチ付きのX1Fかターボでなければなりませんが、元祖X1でも強力なコピー・ツールとして使えます。

また、ロードののち、電源を切らずにIPLに戻ることができれば、元祖X1でもメモリの内容をそっくりテープに落としてしまうことができます。

プロテクトの正面突破に失敗した時でも、後でメモリの中身を解析すれば良いのです。IPLスイッチの付け方はOh! MZ"'83年12月号に紹介されていますので参考にしてください。

●プログラムの入力

さて、先ほどの"64K Loader Saver"を起動し、リスト22を打ち込んで次のようにセーブしてください。

＊SFFC0　FFFF　0000：64K L&S 2

P.111のプログラムから打ち込んだ方は、SHIFT＋BREAKしてSAVEM" CAS：64K L&S 2"、&HFFC0、&HFFFF⏎としてください。その後、64K Loader Saverをロードし、64K L&S2のあとにセーブすればOKです。

この使いかたは、まずゲームなどをロードしたらSHIFT＋BREAKしながらIPLボタンを押してください。(ディスクなどはすぐ抜く）IPLに戻ったら64K Loader Saver IIをロードすると、メイン・メモリの内容が仮想メモリに転送され64K Lorder Saverが起動します。

この状態でダンプなどを取れば、IPLボタンを押す直前のメモリ状態をゆっくりのぞくことができます。後はセーブしようが改造しようが自由にできまｰ。その他の使いかたは64K Loader Saverと全く同じです。

どこからどこまでセーブしたら良いのか分からない場合は、

＊S0000　FEFF〔オートスタート〕

とします。オートスタートを0000か0100または0103にすればほとんどのソフトがIPLロードでスタートするはずです。

初版では85になっているので注意

ソース

```
 0000                     1         OFFSET 0AFC0H-0FFC0H
 FFC0                     2         ORG     0FFC0H
 FFC0                     3 ;
 FFC0 ED 79               4 START:  OUT   (C),A
 FFC2 01 00 40            5         LD    BC,4000H
 FFC5 0A                  6 EXLP:   LD    A,(BC)
 FFC6 ED 79               7         OUT   (C),A
 FFC8 03                  8         INC   BC
 FFC9 78                  9         LD    A,B
 FFCA B1                 10         OR    C
 FFCB 20 F8              11         JR    NZ,EXLP
 FFCD                    12 ;
 FFCD 11 FF 5F           13         LD    DE,5FFFH  ;HL=1FFFH
 FFD0 01 00 40           14         LD    BC,4000H
 FFD3 ED B8              15         LDDR
 FFD5                    16 ;
 FFD5 3E 1D              17         LD    A,1DH
 FFD7 D3 00              18         OUT   (0),A
 FFD9 C3 5B 00           19         JP    005BH
 FFDC                    20 ;
 FFF8                    21         ORG   0FFF8H
 FFF8 FF 3F 00           22         DB 0FFH,3FH,0,01EH,0C0H,0FFH
 FFFB 1E C0 FF
 FFFE                    23 ;
```

<リスト22>

```
ｱﾄﾞﾚｽ +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  CSUM  [  ASCII CODE  ]
-------------------------------------------------------------------------------
:FF00=01 41 33 46 30 30 3D 2B 30 20 2B 31 20 2B 32 20  :CC   .A3F00=+0 +1 +2
:FF10=20 30 20 33 44 20 32 42 20 33 30 20 32 30 20 32  :D2    0 3D 2B 30 20 2
:FF20=30 20 32 30 20 33 32 20 33 30 20 32 30 20 33 32  :C1   0 20 32 30 20 32
:FF30=20 2B 45 20 2B 46 20 20 43 53 55 4D 20 20 5B 20  :54    +E +F  CSUM  [
:FF40=20 41 53 43 49 49 20 43 4F 44 45 20 20 5D 00 00  :61    ASCII CODE  ]..
:FF50=00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F1   ................
:FF60=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0   ................
:FF70=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0   ................
:FF80=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0   ................
:FF90=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0   ................
:FFA0=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0   ................
:FFB0=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0   ................
:FFC0=ED 79 01 00 40 0A ED 79 03 78 B1 20 F8 11 FF 5F  :CA   .y..@..y.xｱ ...
:FFD0=01 00 40 ED B8 3E 1D D3 00 C3 5B 00 FF FF FF FF  :2E   ..@.ｸ>.ﾓ.ﾃ[.....
:FFE0=FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  :F0   ................
:FFF0=FF FF FF FF FF FF FF FF FF 3F 00 1E C0 FF FF FF  :11   .........?..ﾀ...
-------------------------------------------------------------------------------
CSUM  77 6D 55 F0 F7 51 E2 33 0F 8C 19 26 71 FF D5 F9   9E
```

ここのチェックサムが合っていればよい。

この部分だけ入力すればよい

（注　マシン語入力しますかの問いにNと答えてもマシン語入力ができます）

X1用テープ⇨ディスク変換ツール
ディスク・ファイリング・ユーティリティ

《エンゼルソフト》



X1にもディスクが普及してきましたが、いま持っているテープ版ソフトを何とかディスクに落としたいと思っている方も多いのではないでしょうか。

そこでX1版のテープ⇨ディスク変換ツール〝ディスク・ファイリング・ユーティリティ〟を紹介します。

このユーティリティは残念ながらどんなソフトでもディスクに落とせるというわけではありませんが、

1．64Kバイトまでの長いプログラムもディスクに転送できる。(BASICシステムと重なっても大丈夫)

2．テープからディスクだけでなく、ディスク⇨テープ、ディスク⇨ディスクの転送もできる。

3．必要なプログラムをIPL起動にできる。

などの特長がありますので、〝サンダーフォース〟や〝マッピー〟等の超ロング・プログラムまでファイリングできます。プロテクトのかかっているソフトについては64K Loader Saver IIと組み合わせて使えば市販ソフトのほとんどはディスクに落とせるようになるでしょう。

このプログラムはディスクBASIC－CZ8FB01V 1.0 (元祖旧BASIC)でしか動きませんので、X1ターボの方とX1Fの方はシステム・ディスクの中にある Disk Sysgen. Uty によってCZ－8FB 01V1. 0のシステム・ディスクをまず作ってください。

次にリスト1、2、3を打ち込み、それぞれを、

① SAVE〝Start up. Bas〟⏎

② SAVE〝Diskconverter. V-2〟⏎

③ SAVEM〝Loader. 2〟、&HBF38、&HBFFF ⏎

以上の方法でディスクにセーブします。(ファイル・ネームは間違えないようにしてください)これでマスター・ディスクの準備が終わりました。なおリスト中の〝→〟や〝＾z〟などの文字はPOKE&H01A2、&HB7⏎として［ESC］・［CTRL］＋Zのようにすれば表示できます。

●使い方

《テープからディスクへの転送》

①ゲーム・マスターは必ずログイン・ディスク・ドライブにセットしてください。

② RUN〝Start up. Bas〟として〝Start up. Bas〟を起動します。

図1のようなメニューが表示されますが、ここでは[ESC]キーを押してください。

そうすると、図2のように表示されます。(もう一度[ESC]キーを押すとMENUに戻ります)

③転送先のディスク・ドライブNo.を指定します。(そのまま[Return]キーを押すとドライブ0を指定したことになる)

注：ディスク・ドライブが1

〈リスト1〉 Start up・Bas

```
10 '------------------------------------
20 '
30 '   * Disk Filing Utility *
40 '   Copyright 1984 by Angel Soft
50 '          Produced by   Sankodo
60 '
70 '------------------------------------
80 KEY1,"WIDTH 80:FILES"+CHR$(13):CLR:CLEAR&HBF38:MAXFILES2:DEFUSR0=&HBFC8:ONERRORGOTO600:CONSOLE:WIDTH40:DIMFIL$(28):KB
UFOFF:OPTIONSCREEN2:SCREEN
90 CLS:CSIZE2:CURSOR6,0:COLOR2:PRINT#0,"♦♦♦♦";:COLOR5:PRINT#0," MENU ";:COLOR2:PRINT#0,"♦♦♦♦":CSIZE0:J=&HBF38:FORI=0TO19
9:SC=SC+PEEK(I+J):NEXT:IFSC<>18180:LOADM"Loader.2"
100 CLR:DVN=PEEK(&H74BE)AND3:DV$=STR$(DVN)+":":DIM FIL$(28):CURSOR12,2:COLOR7:PRINT" DRIVE (0-3)";DV$;"2++"
110 F=0:FIN=1:DEVI$DV$,0,FA$,SE$:IFMID$(FA$,1,1)=CHR$(1)ANDMID$(FA$,15,3)="Sys":FIL$(0)=MID$(FA$,1,32):FL=0 ELSEFL=1
120 FORI=0TO10:IFF THEN190
130 DEVI$DV$,1S+I,FA$,SE$:IFASC(MID$(FA$,1,1))AND&H88:F=-1
140 FORJ=0TO7
150 IFJ<4:FIL$(FIN)=MID$(FA$,32*J+1,32):GOTO170
160 FIL$(FIN)=MID$(SE$,32*(J-4)+1,32)
170 IF(ASC(FIL$(FIN))AND7)=1:FIN=FIN+1
180 NEXT
190 NEXT
200 FORI=FL TOFIN-1
210 IFI-FL<14:CURSOR0,I-FL+4 ELSECURSOR20,I-FL-10
220 IFI<10:CURSOR1,I-FL+4
230 PRINT#0,I;MID$(FIL$(I),2,16)
240 NEXT
250 FORI=1TOFIN-1:FIL$(I)=LEFT$(FIL$(I),30)+MKI$(CVI(RIGHT$(FIL$(I),2))*16):NEXT
260 CURSOR2,24:S=S AND3:ONS+1GOTO270,280,290,300
270 PRINT"(Device: ";:CREV1:CFLASH1:PRINT#0,"+:Execute ";:CFLASH:CREV:PRINT#0,"+:Save to TAPE)";:GOTO310
280 PRINT#0,"(Device: +:Execute ";:CFLASH1:CREV1:PRINT#0,"+:Save to TAPE";:CFLASH:CREV:PRINT")";:GOTO310
290 PRINT#0,"(ち:Filing ";:CREV1:CFLASH1:PRINT#0,"+:Copy ";:CFLASH:CREV:PRINT#0,"+:Set Auto Run)   ";:GOTO310
300 PRINT#0,"(ち:Filing +:Copy ";:CFLASH1:CREV1:PRINT#0,"+:Set Auto Run";:CFLASH:CREV:PRINT ")   ";
310 CURSOR2,22:PRINT#0,"Please Select";FL;"to";FIN-1;"or +,+,n:,ち";
320 PRINT"   ++";:K$=INKEY$ (1):IF K$="ち":C=0:GOTO590ELSEIFK$="+":S=S+1ELSEIFK$="+":S=S-1ELSEIFK$="↑" ORK$="↓":S=S+2ELSE
340
330 GOTO260
340 IFVAL(K$)=0 ANDLEFT$(K$,1)<>"0"THEN310ELSEPRINTK$;:L$=INKEY$(1):IFL$="ñ":IKY$=K$ELSEIFVAL(K$)=0 ANDLEFT$(K$,1)<>"0"T
HEN310ELSEPRINTL$;:M$=INKEY$(1):IFM$="ñ":IKY$=K$+L$ELSE310
350 IFINSTR(IKY$,":"):DV$=IKY$+":":A=DEVF(DV$):GOTO100
360 IFVAL(IKY$)>FIN-1 ORVAL(IKY$)<FL THEN310
370 INK=VAL(IKY$):FI$=FIL$(INK):CURSOR0,4:PRINT"2";:CFLASH1:PRINT"Loading ";MID$(FI$,2,16):CFLASH
380 FORI=0TO28:FIL$(I)="":NEXT
390 ONS+1GOTO400,430,430,560
400 MEM$(&HFF00,32)=FI$:POKE&HFF00,1
410 POKE&HFF87,&H80+DVN
420 CALL&HBFAA
430 FR=(ASC(MID$(FI$,19,1))+256*ASC(MID$(FI$,20,1))-1)/&H1000:RECN=CVI(RIGHT$(FI$,2)):MEM$(&H1480,32)=FI$:POKE &H1480,1
440 FORI=0TOINT(FR*16)+1
450 DEVI$DV$,RECN+I,E$,F$
460 IFI<&HC0:DEVO$"MEM:",I,E$,F$ELSEMEM$(I*256,128)=E$:MEM$(I*256+128,128)=F$
470 NEXT
480 CURSOR0,4:COLOR1:PRINT"Loading ";MID$(FI$,2,16)
490 IFS=2THEN580
500 CURSOR0,6:COLOR4:PRINT"SET Destination TAPE on Drive CAS0:":
510 CURSOR0,8:PRINT"Ready Ok ? (Y or N)2+++++++";:K$=INKEY$(1):IFK$="n"ORK$="N" THEN100
520 CURSOR0,10:COLOR6:CFLASH1:PRINT"Saving ";MID$(FI$,2,16):CFLASH
530 A=USR0(A)
540 CURSOR0,10:PRINT"Saving ";MID$(FI$,2,16):CFLASH
550 CURSOR0,12:COLOR5:PRINT"Complete !":FORI=1TO5000:NEXT:GOTO100
560 FI$=FI$+STRING$(96,255):MID$(FI$,1,1)=CHR$(1):MID$(FI$,15,3)="Sys"
570 DEVO$DV$,0,FI$,STRING$(128,255):GOTO100
580 C=-1
590 CHAIN"DiskConverter.V-2"
600 BEEP:IFERL=90:GOSUB680:RESUME80
610 IFERL=130:F=-1:RESUMENEXT
620 IFERL=250:RESUMENEXT
630 IFERL=350:DV$=MEM$(&H7495,3):RESUME
640 IFERL=590:GOSUB680:RESUME
650 IFERL=530:CFLASH:RESUME500
660 IFERR=7:CFLASH:RESUME80
670 ONERRORGOTO0
680 PRINT"ñPrease Set Game Master on Drive";DV$;"ññReady Ok?  Hit Any Key";:A$=INKEY$(1):RETURN
```

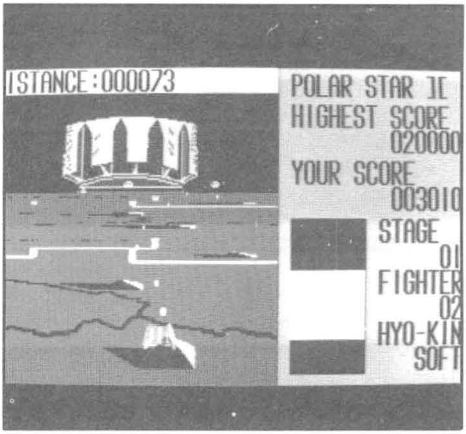
←や＾Ｚなどのコントロール・キャラクタを入れるにはPOKE＆H01A2、＆HB7↙としたあと、[ESC]キーに続いてコントロール・コードのキーを押す。

〈リスト 2〉 Disk Converter. V-2

```
10 '*********************************
20 '******* DISK CONVERTER V2.0 ******
30 '*********************************
40 KBUFON:LIMIT&HBF38:IF C:MEM$(&H1480,32)=FI$:CLR:C=-1:ELSE RUN50
50 ONERROR GOTO1000:DV$=MEM$(&H7495,3):DV=VAL(DV$):IF C:GOTO 70
60 DEFUSR0=&HBF38:DEFUSR1=&HBF68:OPTIONSCREEN 2
70 WIDTH 40:CSIZE 2:COLOR 5:PRINT #0,"** DiskConverter **":CONSOLE 3,22
80 J=&HBF38:FOR I=0TO 199:SC=SC+PEEK(I+J):NEXT:IF NOT(SC=18180OR C):LOADM DV$+"Loader.2"
90 COLOR 4:LOCATE 2,6:PRINT "SET Destination Disk on Drive";DV$;"++";
100 K$=INKEY$(1):PRINT K$:IF K$="ꜫ" THEN 590 ELSE IF K$="ñ" ELSE IF ASC(K$)>47 AND ASC(K$)<52:MID$(DV$,2,1)=K$ ELSE 90
110 IF C THEN 130
120 LOCATE 2,8:PRINT "SET Source TAPE on Drive CAS0:"
130 COLOR 5:LOCATE 6,10:PRINT "Ready Ok?  ꜫ";:COLOR 2:CFLASH1:CREV1:PRINT "Hit Space Key";:CFLASH:CREV
140 K$=INKEY$(1):IF K$="ꜫ"THEN 590 ELSE IF K$=" " OR K$="ñ" ELSE 140
150 COLOR 2:LCCATE 17,10:PRINT "Hit Space Key"
160 COLOR 6:LOCATE 6,12:IF C:PRINT "Saving ";MEM$(&H1481,16) ELSE A=USR0 (1):PRINT "Found  ";MEM$(&H1481,16)
170 IF C THEN 200 ELSE IF MEM$(&H1480,1)=CHR$(1) THEN 200
180 LOCATE 6,14:CFLASH1:PRINT "Irregal File Mode";
190 CFLASH:PRINT"ꜫ":BEEP:BEEP:FOR I=1TO 5000:NEXT:GOTO 90
200 FR=(ASC(MEM$(&H1492,1))+256*ASC(MEM$(&H1493,1))-1)/&H1000:CRST=INT(FR)+1:EDCO=INT(FRAC(FR)*16)+&H80
210 DEVI$ DV$,14,A$,B$:CRSTN=INSTR(A$,STRING$(CRST,0))-1
220 IF CRSTN=-1:LOCATE 6,14:CFLASH1:PRINT"Device Full";:GOTO 190
230 COLOR 7:LOCATE 6,14:PRINT "Is It Right ? (Y or N)+++++++";
240 K$=INKEY$(1):IF K$="N"OR K$="n":CLS:GOTO 90
250 MEM$(&H149D,3)=CHR$(0,CRSTN,0):IF C ELSE MEM$(&H1480,1)=CHR$(1)
260 IF PEEK(&H1481)<&H21:MEM$(&H1481,1)="*"
270 FOR I=2TO 17:IF PEEK(&H1480+I)<&H20:MEM$(&H1480+I,1)=" "
280 NEXT:POKE&H1491,&H20
290 '-- Directry Search --
300 F=0
310 FOR I=0TO 10:IF F THEN 370
320 DEVI$ DV$,16+I,C$,D$
330 FOR J=0TO 7:IF F THEN 360
340 IFJ<4:IF MID$(C$,32*J+1,1)=CHR$(0) OR MID$(C$,32*J+1,1)=CHR$(255):STT=J:RECN=16+I:F=-1:GOTO 360
350 IFJ>=4:IF MID$(D$,32*(J-4)+1,1)=CHR$(0) OR MID$(D$,32*(J-4)+1,1)=CHR$(255):STT=J:RECN=16+I:F=-1
360 NEXT
370 NEXT
380 IF STT<4:MID$(C$,STT*32+1,32)=MEM$(&H1480,32)
390 IF STT>=4:MID$(D$,(STT-4)*32+1,32)=MEM$(&H1480,32)
400 IF C:LOCATE 6,18:COLOR 6:CFLASH1:PRINT "Now Saving !":CFLASH:PRINT "        ":GOTO 440
410 LOCATE 6,16:COLOR 5:CFLASH1:PRINT "Loading "+MEM$(&H1481,16):A=USR1 (A):CFLASH
420 LOCATE 6,16:COLOR 1:PRINT "Loading "+MEM$(&H1481,16)
430 LOCATE 6,18:COLOR 6:CFLASH1:PRINT "Now Converting !":CFLASH:PRINT "        "
440 FOR I=0TO INT(FR*16)+1
450 IF I<&HC0:DEVI$ "MEM:",I,E$,F$ ELSE E$=MEM$(I*256,128):F$=MEM$(I*256+128,128)
460 DEVO$ DV$,CRSTN*16+I,E$,F$
470 NEXT
480 '-- File Alocation Table Write -
490 IF F:DEVO$ DV$,RECN,C$,D$
500 FOR I=1TO (CRST-1)
510 MID$(A$,CRSTN+I,1)=CHR$(CRSTN+I)
520 NEXT
530 MID$(A$,CRSTN+CRST,1)=CHR$(EDCO)
540 IF F:DEVO$ DV$,14,A$,B$
550 LOCATE 6,18:COLOR 4:PRINT "Now Converting !"
560 LOCATE 6,20:COLOR 3:PRINT "Compleate !"
570 LOCATE 6,22:COLOR 7:PRINT "Continue or Execute ? (C or E)+++++++";
580 K$=INKEY$(1):IF K$="E" OR K$="e" THEN 590 ELSE CLS:GOTO 90
590 CLR:CHAIN "Start up.Bas"
1000 DV$=MEM$(&H7495,3):DV=VAL(DV$)
1010 IF ERR=73OR ERL=460:CFLASH1:LOCATE 6,14:IF ERR=73:PRINT"Device Error";:RESUME 190:ELSE PRINT"Write Protected";:CFLA
SH:PRINT"ꜫ":BEEP:BEEP:FOR I=1TO 10000:NEXT:C=-1:RESUME 90
1020 IF ERL=590:PRINT"ñPrease Set Game Master on Drive";DV$;"ññReady Ok? Hit Any Key";:K$=INKEY$(1):RESUME
1030 CFLASH:ONERROR GOTO 0
```



Disk Converterの実行画面。



Diskに落としたポーラースター。

〈リスト3〉 Loader. 2

アドレス	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	CSUM
:BF00=	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
:BF10=	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
:BF20=	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
:BF30=	00	00	00	00	00	00	00	00	21	80	14	01	20	00	CD	41	:E4
:BF40=	00	2A	94	14	7C	3C	B5	20	19	21	00	FF	ED	4B	92	14	:76
:BF50=	CD	44	00	2A	12	FF	22	92	14	2A	14	FF	22	94	14	22	:3D
:BF60=	96	14	3E	01	CD	EC	0D	C9	21	15	0C	36	CD	23	36	92	:A8
:BF70=	23	36	BF	21	00	00	ED	4B	92	14	CD	44	00	21	15	0C	:6A
:BF80=	36	77	23	36	23	23	36	0B	3E	01	D4	EC	0D	D0	3E	1D	:C4
:BF90=	DD	E9	08	7C	FE	C0	38	04	08	77	18	0B	E5	D9	C1	78	:DD
:BFA0=	C6	40	47	08	ED	79	D9	23	0B	C9	31	00	00	F3	2E	00	:DD
:BFB0=	CD	33	01	01	00	1D	ED	79	21	F5	00	E5	3E	07	32	86	:7D
:BFC0=	FF	C3	D2	01	00	00	00	00	21	80	14	01	20	00	CD	3B	:73
:BFD0=	00	21	E9	BF	22	C4	0B	21	00	00	ED	4B	92	14	D4	3E	:CB
:BFE0=	00	21	8A	0C	22	C4	0B	18	9F	7C	FE	C0	38	03	7E	18	:6A
:BFF0=	0A	E5	D9	C1	78	C6	40	47	ED	78	D9	CD	8A	0C	C9	00	:B8
CSUM	35	75	22	A8	25	EE	5B	F1	20	9E	F6	2E	A0	E9	05	C1	04

〈図2〉

```
         ◆◆◆◆ MENU ◆◆◆◆
          DRIVE(0-3) 0:

 0 BASIC CZ8FB01 Sys
 1 BASIC CZ8FB01 Sys
 2 Loader
                   2
 Pleae Select 0 to 2 or ←,→, n:,Ec
(Device: ←:Execute →:Save to TAPE)
```

〈図1〉

```
    ** DiskConverter **
SET DistinationDisk on Drive0:
```

```
   Execute    ⇔   Save to TAPE
      ↑               ↑
      ↓               ↓
   Set Auto Run  ⇔   Copy
```

〈図3〉

台のみの方は、セーブするディスクをセットし［Return］キーを押してください。
④ディスクに転送したいプログラム・テープをデータコーダにセットし［SpaceBar］を押してください。
⑤データレコーダがいったん止って、File Name を表示します。正しければ［Y］キーを押してください。
⑥読み込みが始まり、その後自動的にディスクに転送されます。続けて転送する場合は［C］キーを押してください。

《MENU の使い方》

"Start up. Bas"を起動すると、テープからディスクへの転送でも説明したようにMENUが表われます。

MENU には次の機能があります。

1．ディスクに登録されたプログラムの実行
2．ディスクからテープへの転送
3．ディスクからディスクへの転送
4．IPL スタート（Auto Run）の設定

① "Start up. Bas" を起動します。1：［Return］のようにしてゲーム・マスターが入っているドライブNo.を指定します。(最初はドライブ0にセット)

注：ディスク・ドライブが1台のみの場合はセーブするディスクをセットし［Return］キーを押してください)
②ディスクに登録されているプログラムのFile - Nameが表示されます。この状態でカーソル・コントロール・キー（⇦、⇨）を押すと画面表示下部では図3のように変わります。(最初はExecuteにセットされています)

《プログラムの実行》

③カーソル・コントロール・キーで'Execute'を選んでください。
④実行したいプログラムの番号を入力して［Return］キーを押します。(目的のプログラムが見付からなかった時は、1：［Return］のようにドライブを指定してください)

注：ディスク・ドライブが1台のみの場合は目的のプログラムが入ったディスクをセットし、0：［Return］と入力してください。

《ディスクからテープへの転送》

③カーソル・コントロール・キーで'Save to TAPE'を選んでください。
④テープにセーブしたいプログラムの番号を入力して[Return]のキーを押してください。
⑤データレコーダに新しいテープをセットして、[Return]キーを押してください。

《ディスクからディスクへの転送》

③カーソル・コントロール・キーで'Copy'を選んでください。
④ディスクにセーブしたいプログラムの番号を入力して［Return］キーを押してください。

注：ディスク・ドライブが1台の時は、目的のプログラムが入ったディスクをセットし、0：［Return]と入力してください。
⑤転送先となるディスクをセットして[Return]キーを押すか、ドライブNo.を指定します。
⑥準備がOKであれば、[Y]キーを押します。

《IPLスタートのセット》

③カーソル・コントロール・キーで'Set Auto Run'を選びます。
④'Auto Run'をかけたいプログラムの番号を入力して[Return]キーを押します。

注：'Auto Run'をかけたプログラムはMENUで、0番として表示されます。(最初はDisk BASICにセットされています。)

第4章 パソコン周辺機器製作集



ROMソフトがテープに落とせる
強力カバージョンUP版

MSX



バックアップMSX Ver.2の製作

橋本　大

バックアップ活用テクニック①のバックアップMSXver.1を作ったけれども、コピーできないROMがたくさんあるので困った人がいるのではないでしょうか。

また、32Kや64Kのマシンではうまく動かないという人もいるでしょう。これらは、RAMカードの改良とさらにソフトの強化により、かなり改善ができます。そこで、まず最初にゲームROMの種類について調べたことをお話します。なお、本記事と供にバックアップ活用テクニック①をご覧いただければBESTです!!

●ゲームROMの種類

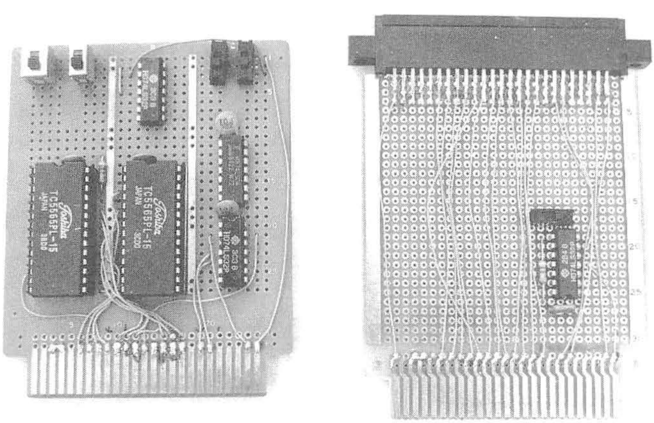
ゲームROMの種類には、

① 32Kバイトのもの
② 16Kバイトのもの

に分けられ、さらに

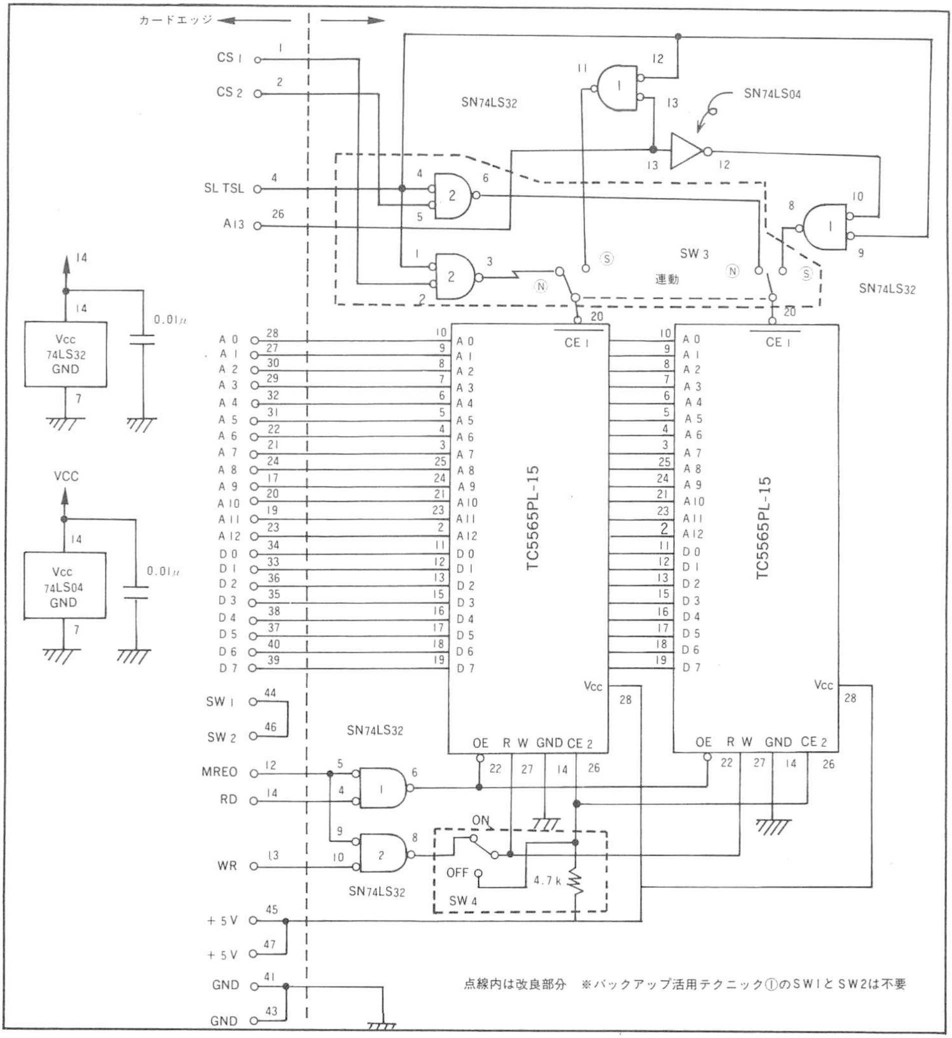
A：RAM上で動作するもの
B：書き込みを禁止しないとダメなもの

また、②にはアドレスが4000H～7FFFHに位置しているものと、8000H～BFFFHに位置しているもの、そして筆者が新発見したものに4000H～5FFFHと8000H～9FFFHに分かれている"NAMCOTタイプ"(以下、Nタイプと略す)があります。これはナムコットのROMに多くみられます。さらに、容量は16Kバイトなのに CS12で使う見せか



左が改造したRAMカード。右がカートリッジ・アダプタ。

〈図1〉改良RAMカードの回路



けの32ＫＲＯＭもあります。(以下、Ｔタイプと略す)

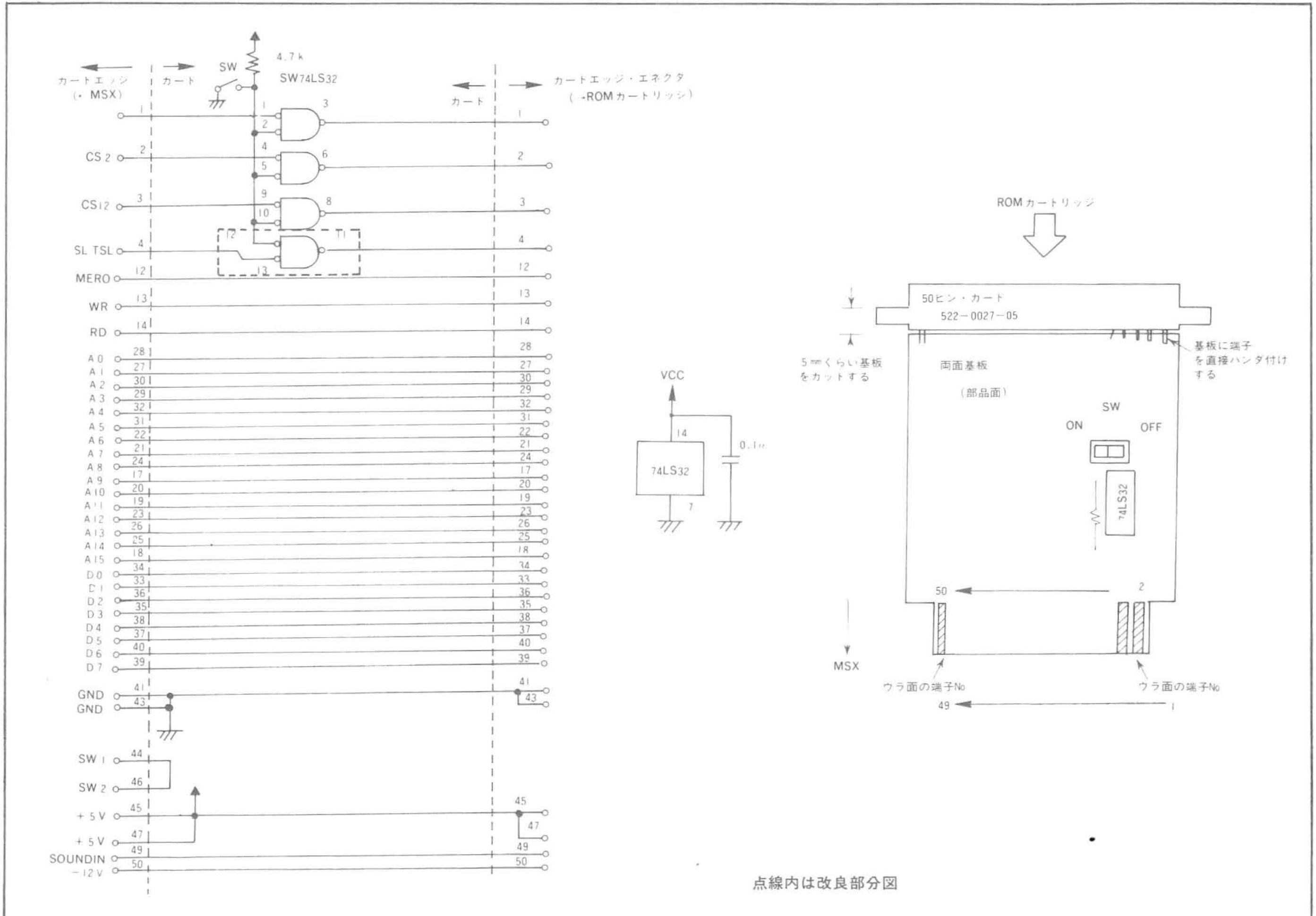
●RAMカードの改良

ＮタイプＲＯＭ対応用にＲＡＭを8000Ｈ～9ＦＦＦＨと4000Ｈ～5ＦＦＦＨに分け、さらにＴタイプに対応させるためにＣＳ12でも使えるようにします。

用意する部品は２回路２接点のスイッチと１回路２接点のスイッチ、それと配線材だけです。ＯＲはＳＮ74ＬＳ32の［２］の方の余っている素子を使います。(図１の回路図を参照) ＳＷ３をＳにすると普通のＲＯＭ対応でＮにするとメモリが分かれてＮＡＭＣＯＴ対応になります。ＳＷ４はＲＡＭへの書き込み許可スイッチで、これをＯＦＦにするとＲＯＭと同様に扱えます。

このＲＡＭカードはＳＷ２をＲＡＭ、ＳＷ

〈図2〉改良カートリッジ・アダプタ回路



点線内は改良部分図

3をSにしてSW4をONにすると普通の増設RAMの代わりになります。また、このままで4000H～7FFFHの裏RAMになります。以下、SW2はRAM側にしSW3、SW4のみを使用します。カートリッジ・アダプタも図2のように改良してください。

●ROMの調べ方

このROMは、どんなタイプか見分けるにはどうしたら良いのか？その方法は？

まず、ROMの4000Hが41Hで、4001Hが42Hの値になっていれば①か②の4000H～7FFHまたは②のNタイプまたは②のTタイプです。

内容を見て8000H番地以上をCALLしていれば（例えばC3□□△□とかCD□□△□やC2□□△□、CA□□△□とある場合：□の中は適当な数字、△は8H～BH）①または②のNタイプまたはTタイプです。

8000Hが41Hで8001Hが42Hのときは②の8000H～BFFFHのものです。

②のNタイプの場合は4000H～5FFFHと6000H～7FFFHの内容が、そっくり同じになっています。Tタイプはソフト的に見ると32KバイトROMのようですが、分析してみるとROMは16KバイトでCS12につながっています。走らせてみないと分かりませんが、ほとんど32Kに見られるのはこのTタイプです。なお、32KのROMを使っているゲーム・カートリッジは価格が高いので見分けがつきます。

AかBかは走らせてみないと分かりませんが、Aの場合ならオートスタートさせることが可能ですし、64KのマシンならRAMカードなしで実行可能です。

部品表

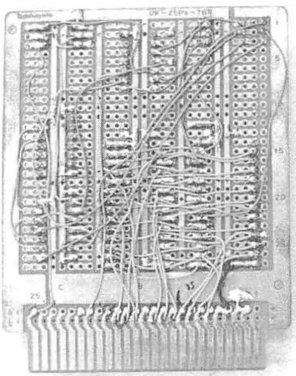
●カートリッジ・アダプタ

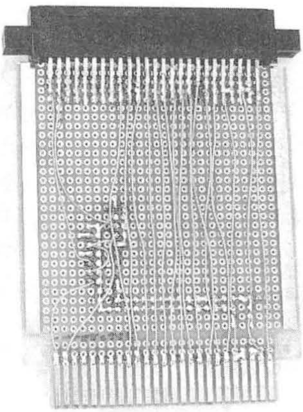
		参考価格@
基板　サンハヤトMCC－158	……1	1400円
50ピン　カードエッジ・コネクタ	…1	300円
SN74LS32	……1	？
SW（1回路2接点）	……1	70円
抵抗4.7KΩ	……1	10円
コンデンサ0.1μF	……1	20円

●RAMカートリッジ

		参考価格@
基板　サンハヤトUK－25PA－76R	……1	1000円
TC5565PL－15（8KバイトRAM）	……2	1900円
SN74LS32N	……2	？
SN74LS04N	……1	？
28ピンICソケット	……2	100円
コンデンサ0.01μF	……4	20円
SW（1回路2接点）　※新規に製作する場合は1個でも可	……3	70円
SW（2回路2接点）	……1	100円
抵抗4.7KΩ	……1	10円

※基板とコネクタとICはアド電子。その他はラジオデパート。TC5565PL－15はヒロセパーツセンターなら2200円。（東京秋葉原にて）





RAMカード、カートリッジ・アダプタの配線

●マシンの制限

ROMをコピーして、それを実行するためには16Kバイト以上のRAMがあり、スロットが2つ以上必要ですが、32Kのマシンや64Kのマシンはスロットは1つでOKです。RAMカードの容量が今回、16Kバイトなので②のROMしかコピーできません。32KのバイトのプログラムをコピーするにはRAMカード2枚が必要です。

●バックアップMSXver.2プログラムについて

リスト1のプログラムは、スロットの内容をRAM上の9000H～CFFFHに転送してその内容をテープにコピーします。スロットを1以外で行ないたい場合は、DATA文の中の3E01を全部3E0□としてください。(□の中は使いたいスロットのナンバー)

このプログラムは4000H～のものか8000H～のものかNタイプのものかは自動的に判断しますが、それが必ず合っているとは限りませんので、マニュアルでも選べるようにしました。(NタイプでもNタイプと判断されないROMの可能性があるから)

32KバイトROMの場合とTタイプの場合は、4000H～のものと判断されるので注意してください。(Tタイプのものは4000H～のものと同様に扱う)

また、もし使っているデータレコーダが2400ボー対応でない場合は、プログラムの10行をSCREEN1,,, 1にしてください。これでセーブを1200ボーで行ないます。

●コピーの仕方

次にコピーの仕方を順序だてて紹介しましょう。

[1] 16Kのマシンは増設RAMをスロット2に入れる。カートリッジ・アダプタのスイッチをOFFにして、スロット1に入れる。もちろん、ROMを付けておく。(RAMカードを使うときは、SW1：CS2でもCS1でも良い。 SW：2RAM、SW3：S、SW4：ONにしてスロット2に入れる)

[2] 電源を入れてリスト1のプログラムを走らせる。(メニューがでたなら通常はAUTOにする)

もし ROM ERROR がでたならスイッチがOFFのままか、ROMが差し込んでないか、接点不良ですから初めからやり直してください。何度やってもダメなときは、機器の故障かプログラムのミスです。

COPY ERRORが出たときはRAMの異状です。もう一度やり直してください。

VERIFY ERRORはROMの内容がテープにうまくコピーできなかったということです。レベルの調整をしてもう一度トライしてみてください。(表1参照)

●コピーしたプログラムの走らせ方

以下、SW1はどちらにセットしても良い。SW2は必ずRAM側にする。SW1、SW2がない場合は関係ない。

16Kマシン使用時

★Nタイプ以外の場合

①RAMカードを入れる。SW3：S、SW4：ON

②電源を入れる。

③BLOAD "CAS：", Rを実行

④SW4：OFF

⑤リセットまたはDEFUSR(0)=0：I=USR(0)を実行

★Nタイプの場合

①～③は同上

④SW3：N

⑤リセットまたはDEFUSR(0)=0：I=USR(0)を実行

〈リスト1〉 バックアップMSXver.2プログラムリスト

```
10 SCREEN 1,,,2
20 CLEAR 100,&H9000
30 S=&HF000:E=&HF0D7:GOSUB100
40 S=&HD000:E=&HD01F:GOSUB100
50 S=&HE000:E=&HE017:GOSUB100
60 GOTO 170
70 '
80 ')))))))))))) ﾏｼﾝｺﾞ ｶｷｺﾐ (((((((
90 '
100 FOR J=S TO E
110 READ B$
120 POKE J,VAL("&H"+B$)
130 NEXT:RETURN
140 '
150 '))))))))))) ｼｮｷ ｾｯﾃｲ (((((((((
160 '
170 PRINT"ROMｱﾀﾞﾌﾟﾀ ﾉ SW ｦ ON ﾆｽﾙ｡"
180 I$=INPUT$(1)
190 DEF USR=&HE000:I=USR(0)
200 IF PEEK(&HF300)= 65 AND PEEK(&HF301)=66 THEN PRINT"ｺﾚﾊ 4000H ﾀｲﾌﾟ ﾉ ROM ﾃﾞｽ"
:R=0:GOTO 280
210 POKE &HE002,&H80:POKE &HE00D,&H80
220 I=USR(0)
230 IF PEEK(&HF300)= 65 AND PEEK(&HF301)=66 THEN PRINT"ｺﾚﾊ 8000H ﾀｲﾌﾟ ﾉ ROM ﾃﾞｽ"
:R=1:GOTO 280
240 PRINT"ROM ERROR":END
250 '
260 '))))))))))) ｾﾚｸﾄ (((((((((((((((
270 '
280 INPUT"0-AUTO   1-4000H ROM 2-N ﾀｲﾌﾟ3-VERIFY 4-8000H ROM   ";A
290 IF A<0 OR A>4 THEN 280
300 IF A=3 THEN 570
310 IF A=4 OR R=1 THEN 700
320 IF R=1 THEN PRINT"ROM ERROR":END
330 IF A=2 THEN R=1
340 '
350 '))))))))))) 4000H & NAMCOT (((((((
360 '
370 POKE &HF100,&HFF
380 R=R*&H30
390 DEF USR=&HF000+R:I=USR(0)
400 DEF USR=&HF050+R:I=USR(0)
410 IF PEEK(&HF100)<>0 THEN PRINT" COPY ERROR":END
420 IF R<>0 THEN 450
430 POKE &HF100,&HFF:DEF USR=&HF0A8:I=USR(0)
440 IF PEEK(&HF100)=0 AND A=0 THEN PRINT"ｺﾚﾊ NAMCOT ﾀｲﾌﾟ ﾃﾞｽ ":R=1:GOTO 380
450 '
460 '))))))))))) SAVE ((((((((((((((((
470 '
480 PRINT"SAVE ﾉ ﾖｳｲ ｦ ｼﾃｸﾀﾞｻｲ"
490 F$="":INPUT"FILE NAME";F$
500 PRINT"S KEY ﾃﾞ SAVE ｼﾏｽ":A$=INPUT$(1):IFA$<>"S" AND A$<>"s" THEN 500
510 BSAVE"CAS:"+F$,&H9000,&HD01F,&HD000
520 INPUT "ﾓｳ ｲﾁﾄﾞ SAVE ｼﾏｽｶ ";A$:IF A$="Y" OR A$="y" THEN 500
530 END
540 '
550 '))))))))))) VERIFY (((((((((((((((
560 '
570 INPUT"1-4000H ROM  2-NAMCOT ﾀｲﾌﾟ 3-8000H ROM ";A
575 POKE &HF100,99
580 IF A<1 OR A>3  THEN 570
590 PRINT"LOAD ﾉ ﾖｳｲ ｦ ｼﾃｸﾀﾞｻｲ"
```

```
600 PRINT"L KEY デ" LOAD シマス":IF INSTR("1L",INPUT $(1))=0 THEN 600
610 BLOAD"CAS:"
620 A=A-1:IF A=1 THEN A=&H30
630 IF A=2 THEN A=&H78:POKE &HF0D2,&H59
640 DEF USR=&HF050+A:I=USR(0)
650 IF PEEK(&HF100)<>0 THEN PRINT" VERIFY ERROR"
660 END
670 '
680 ')))))))))) 8000H ROM ((((((((((
690 '
700 DEF USR=&HF0C8:I=USR(0)
710 POKE &HF100,&HFF
720 POKE &HF0D2,&H59
730 DEF USR=&HF0C8:I=USR(0)
740 IF PEEK(&HF100)<>0 THEN PRINT" COPY ERROR":END
750 GOTO 480
760 '
770 ')))))))))))コピー((((((((((((
780 '
790 DATA 21,00,40,01,00,50,11,00
800 DATA 40,3E,01,D5,C5,CD,0C,00
810 DATA 00,00,C1,E5,09,77,E1,D1
820 DATA 23,1B,7A,B3,C2,09,F0,C9
830 DATA 00,00,00,00,00,00,00,00
840 DATA 0C,00,00,00,00,00,00,00
850 DATA 21,00,40,01,00,50,11,00
860 DATA 20,CD,09,F0,00,00,00,00
870 DATA 21,00,80,01,00,30,11,00
880 DATA 20,CD,09,F0,C9,00,00,00
890 DATA 21,00,40,01,00,50,11,00
900 DATA 40,D5,C5,E5,09,56,E1,D5
910 DATA 3E,01,CD,0C,00,D1,BA,C1
920 DATA D1,C0,23,1B,7A,B3,C2,59
930 DATA F0,AF,32,00,F1,C9,00,00
940 DATA 00,00,00,00,00,00,00,00
950 DATA 21,00,40,01,00,50,11,00
960 DATA 20,CD,59,F0,3A,00,F1,B7
970 DATA C0,3E,FF,32,00,F1,00,00
980 DATA 21,00,80,01,00,30,11,00
990 DATA 20,CD,59,F0,C9,00,00,00
1000 DATA 21,00,40,01,00,70,11,00
1010 DATA 20,CD,59,F0,C9,00,00,00
1020 DATA 21,00,40,01,00,50,11,00
1030 DATA 40,CD,09,F0,C9,00,00,00
1040 DATA 21,00,80,01,00,10,11,00
1050 DATA 40,CD,09,F0,C9,00,00,00
1060 '
1070 ')))))))))) START ((((((((((
1080 '
1090 DATA 3E,01,21,00,40,CD,24,00
1100 DATA 21,00,90,01,00,40,11,00
1110 DATA 40,ED,B0,21,00,40,3E,00
1120 DATA CD,24,00,00,00,C9,00,00
1130 '
1140 '))))))))) ROM CHECK ((((((
1150 '
1160 DATA 21,00,40,3E,01,CD,0C,00
1170 DATA 32,00,F3,21,01,40,3E,01
1180 DATA CD,0C,00,32,01,F3,C9,00
```

〈表1〉 コピー手順（ROM→テープ）

プログラムをロードする
↓
RUN

※ROMがNタイプの時は、表示されるので忘れずに記憶しておく。

ROMのSWをON
↓
0を入力
↓
SAVE

VERIFY
↓
END

※信頼できるテープレコーダならVERIFYを省略しても可。

エラー表示の内容
①ROM ERROR……ROMの内容が読めない。ゲームROMではない等が原因。
②COPY ERROR……RAMに書けない。
③VERIFY ERROR……テープ内容とROMの内容が違う。

※①②はカートリッジの接触不良が多い。RAMカードの故障も考えられる。

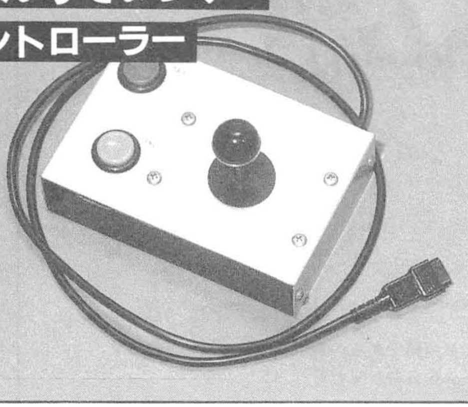
32Kと64Kマシン使用時

★Nタイプ以外
①RAMカードを入れる。SW3：S、SW4：OFF
②電源を入れる。SW4：ON
③BLOAD〝CAS〞，Rを実行
④SW4：OFF
⑤リセットまたはDEF USR（0）=0：I=USR（0）を実行。
★Nタイプの場合
①～③は同上
④SW3：N
⑤リセットまたはDEF USR（0）=0：I=USR（0）を実行。

P.S.「風の谷のナウシカ」は、とっても面白いからビデオで皆んな見るように!!（感動もの）

〈表2〉 バックアップ・テスト

ソフト名	コピー	タイプ
コナミのゴルフ	OK	4000H
イーアルカンフー	OK	4000H
王家の谷	OK	4000H
ギャラガ	OK	N
ボスコニアン	OK	N
スーパーシンセ	OK	4000H
ラリーX	OK	N
マッピー	OK	N
ボコスカウォーズ	OK	8000H
SASA	OK	8000H
FRONT LINE	OK	T
A. E	OK	4000H
ピットフォール	OK	T
エクセリオン	OK	4000H
熱戦甲子園	OK	8000H
野球狂	NO	32Kと思われる



PC-8801mkII/SR

mkIIの登場!

なんとターボ付き

櫟木義広

さてさて、バックアップ活用マニュアル①で好評だった〝じょいすていっくコントローラー〟の上位バージョン、名付けて「じょいすていっくコントローラー mk II」の登場だヨーン!!

バックアップマニュアル①ではたくさんの人に喜んでもらえたようで、どうもありがとうございます。私と同じ気持ちの人が多勢いるようで、作られた方々も多いと思います。が、最近、他のゲームを入手して遊んでいたところ、いかんせん機能に不満が出て来ちゃいまして（自分で作った作品のくせに、このいいかげんさは何であろうか……）前回のI号機にプラスアルファという形で機能のアップを計る事にしました。I号機を作った人でも少々の部品の追加で済みますので、バージョンアップすることをおススメします。

●mkIIとは

改造はトリガー・スイッチ周辺に対し行ないました。I号機ではトリガースイッチの切り替えは2種類のみでしたが、対応しきれない事を痛感し、トリガー・スイッチ1、2それぞれ別々に切り替えられるようにし、対応するキーもトリガー1についてはSPACE、X、ESC。トリガー2つではRETURN、Z、SHIFTと増やし、柔軟な対応を可能とさせました。そしてもうひとつ、本書のファミコンの記事にもあるように、今流行のTurbo、たーぼを装着しました。

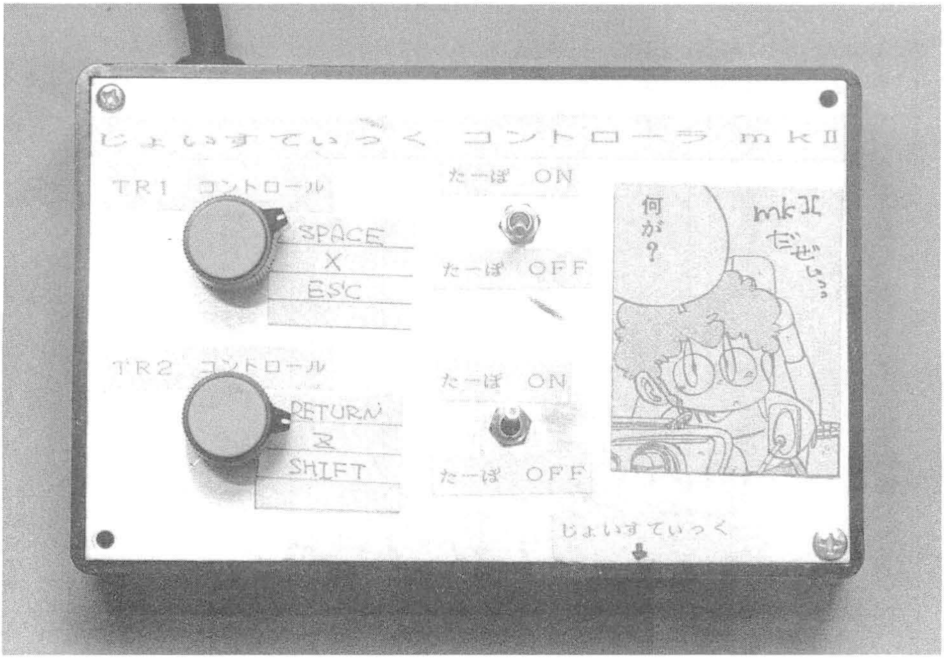
これもトリガー・スイッチの1、2共に別々に作り、独立してたーぼのON、OFFができるようにしてあります。

以上の機能を実現させるために、ICを総計5コ使います。ジョイステック、及びトリガー・スイッチの状態をキーボード・データとして出力するためにデコーダの74159、アナクロ、違った、アナログ・スイッチを使用します。

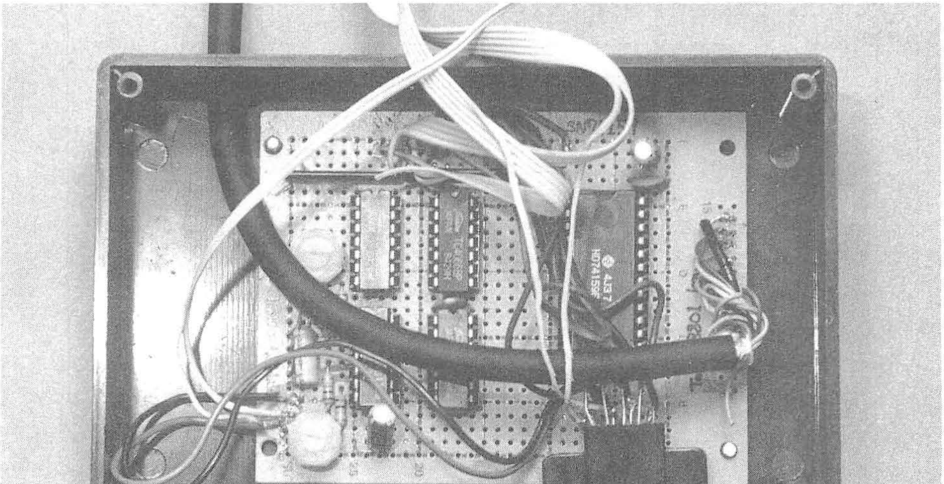
そして、たーぼ回路にはHKS・Turboを、いやいや違う、4011を使用しています。たーぼのスピードは4011のVR1、VR2でそれぞれトリガー・スイッチ1、2のコントロールをしていますので、この2つのVRの調整で速いのから遅いのまで変化させる事ができますので、各自好みの、というよりゲームに適したスピードで使用するようにしてください。

●mkIIのパーツ

mK IIで使うパーツで時殊なものは88のキーボード・コネクタの13ピンのDINプラグだけでしょう。

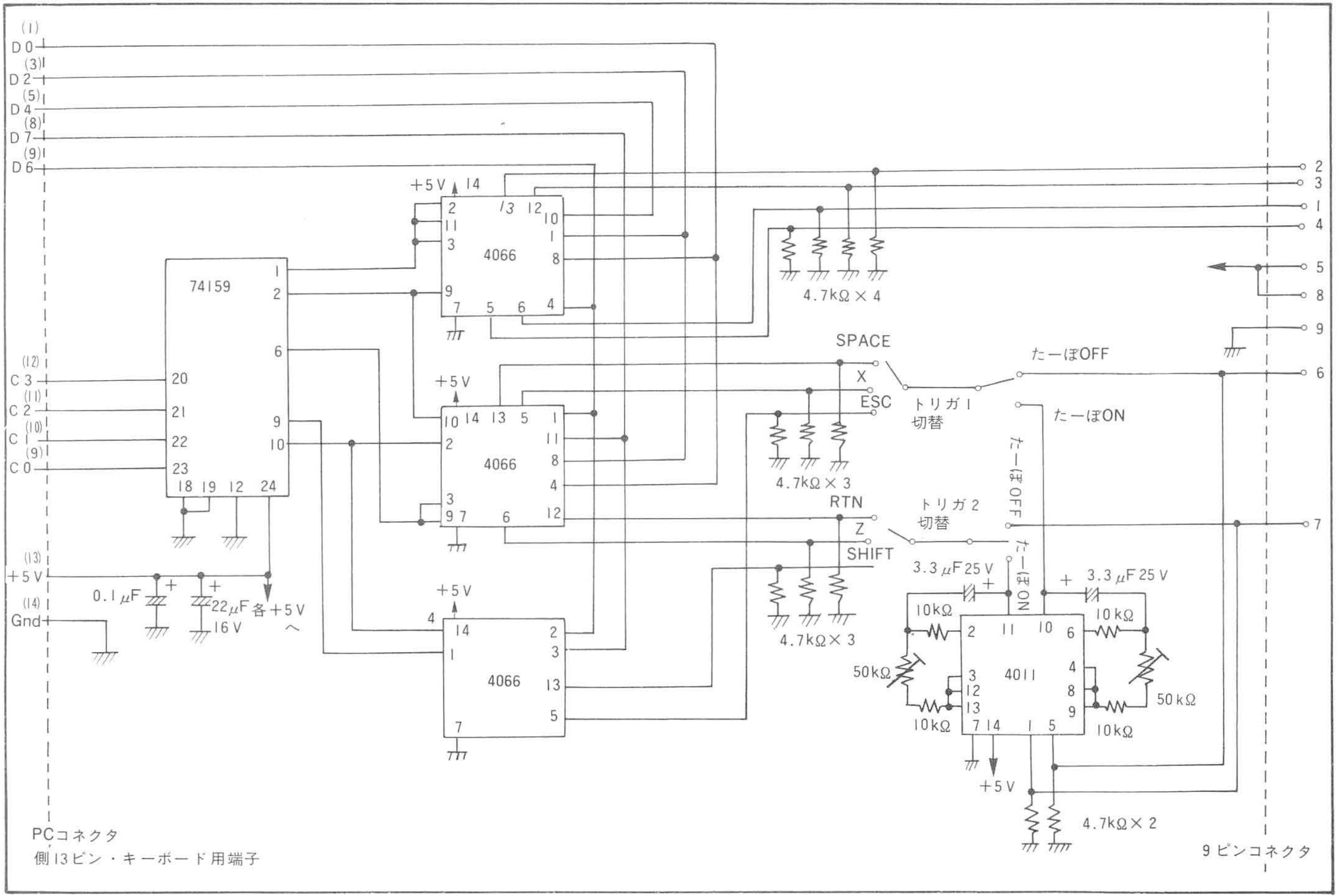


できあがったコントローラーmkⅡのパネル。



コントロール部の配線。

〈図1〉コントローラmkIIの回路図



なかなか見つけられませんが、前回でも紹介しました秋葉原の「ミマツ音響㈱」で入手できますし、時々ジャンク屋にも顔を出すようです。その他はもうかなり一般的になってきた部品ばかりですのですぐ集まるでしょう。ただ、ケースだけは全てのパーツが収まる大きさのを用意してください。(当り前だ‼ってか？)

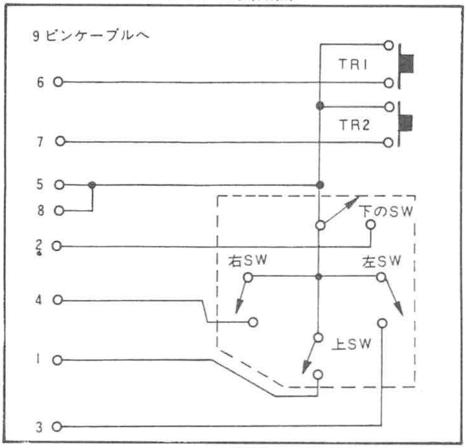
したがってケースの加工図はあくまでも私と同等のパーツがあるという前提のものですので、くれぐれも気を付けて。

製作上の注意点を申し上げます。これは重要な事です。間違えないでください。(う〜ん鋭い指摘だ‼)部品をこわさないでください。(なんて私は頭がいいのだろう？　こんなわかりやすい説明はないなぁ) これだけです。

●mkIIを使おう

I号機もそうですが、コントローラーのジョイスティック・コネクタは、市販のMSX用

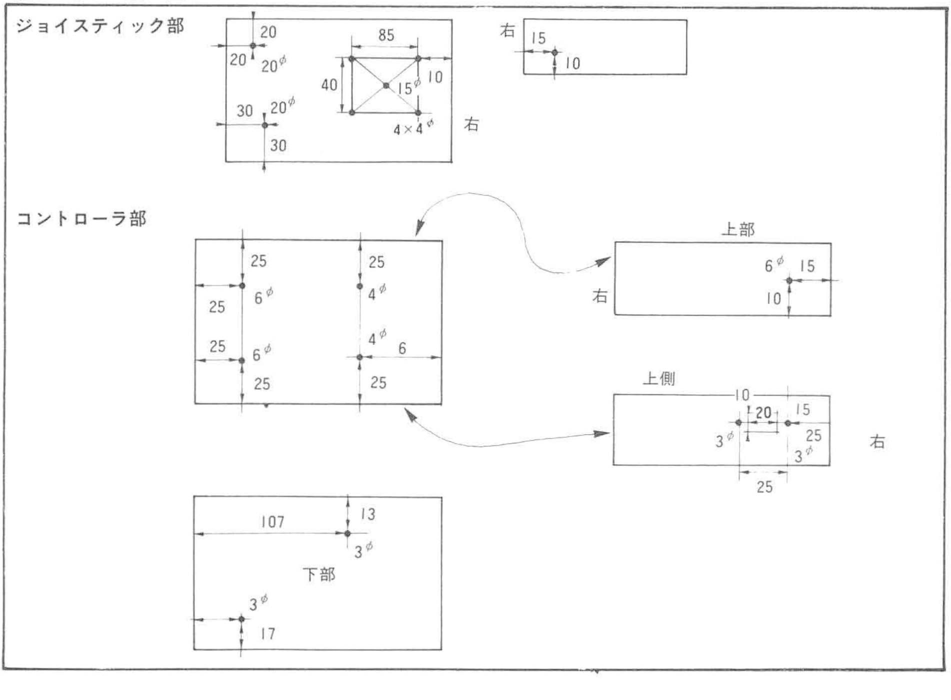
〈図2〉ジョイスティックの回路図



などのジョイスティックに合わせてあるので、既にMSX用のカッコイイのを持っているのならば、そのまま使用できます。

使い方などはいまさらいう事も無いでしょうが、コントローラの13ピンコネクタを88の2つあるキーボード・コネクタのどちらか空

〈図3〉ケース加工図



いている方へ差し込み、ゲームの説明書通りに使うキーをトリガー・スイッチに割り当てて遊ぶだけです。

注意してほしい事があります。ジョイスティックを動かした場合、本機ではテン・キーの2、4、6、8に対応していますので、スティックを斜めに動かした場合は例えば左上に倒した時はテンキーの4と8を押した事になります。ある種のゲームでは、例えば左上に動く時に7のキーを押さないとダメなようなものは動きませんので注意してください。

それと、たーぼスイッチを入れた方がいい時と入れない方がいい時とありますので、ゲームによって各自いじってみて下さい。

●mkIIの次はSR!?

mK IIなんてものを作ってしまいましたが、みなさんも各自研究してより使い易いものを作るなりしてくださいね。(うーん mK IIの次はSRか?それともZ (ゼータ) かな?) いいのが出来たら編集部にでも教えてください!!

DO!! チャレンジですよ!! YLのファンレターをお待ちしております!!

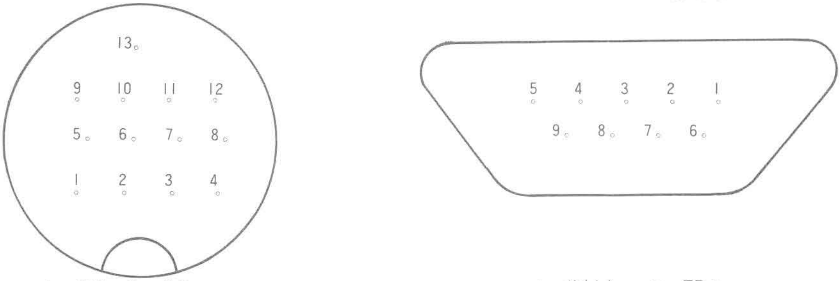
部品表

4方向(8方向)ジョイスティック……1
押しボタン・スイッチ……2
IC CD4066 ……3
CD4011 ……1
SN74159 ……1
14ピンICソケット……4
24ピンICソケット……1
1ツ目穴あき基板 100mm×70mm位 ……1
ケース ジョイスティック部, タカチYM-150 ……1
コントローラ部, 共立UB-8 ……1
ロータリ・スイッチ, アルプス製2回路5接点……2
トグル・スイッチ 3ピン ……2
ツマミ……2
9ピンコネクタ……1組
13ピンPC88キーボード用コネクタ ……1
抵抗
4.7kΩ 12本または集合抵抗2つ
10kΩ……4
50kΩ半固定……2
コンデンサ
22μF 16V……1
3.3μF 25V……2
0.1μF……1
ゴムブッシング……1
電線……少々
3×10mm ビス・ナット……〃
4×10mm ビス・ナッス……〃

〈図4〉コネクタへの配線

13ピンDINコネクタ
プラグのハンダ付け面より

9ピン・コネクタ
コネクタのハンダ付け面より

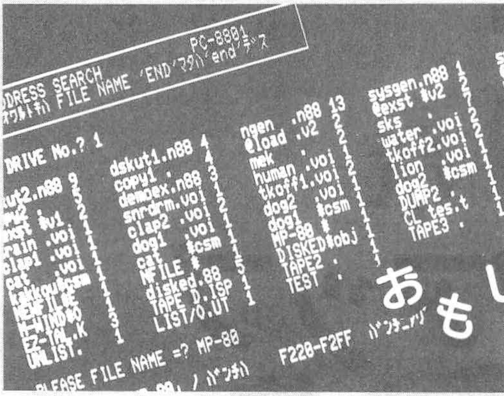


1−D0 8−D7
2−D1 9−C0
3−D2 10−C1
4−D3 11−C2
5−D4 12−C3
6−D5 13−+5V
7−D6 14−Gnd

1−前(上) 7−TR2
2−後(下) 8−コモン
3−左 9−Gnd
4−右
5−+5V
6−TR1

第5章
シェイプアッププログラム集





PC8001/mkII/SR

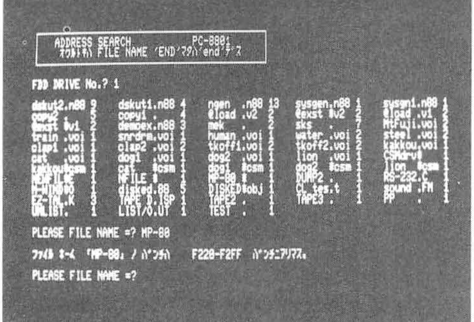
PC-8801 おもしろプログラム集

岡本伸一

このコーナーでは、4つの短いながらも使って便利なプログラムを紹介します。キーボードになれている方ならすぐにINPUTできる短いプログラムですので是非試して見てはいかがですか。?

●アドレス・サーチ・プログラム

最初に紹介するプログラムはアドレス・サーチ・プログラムです。このプログラムは、ゲームソフトなどのマシン語プログラムのメモリ・エリアを知りたい時に便利なプログラムです。使い方はアドレス・サーチ・プログラムのリストを間違いのない様にキーボード



アドレス・サーチ・プログラムの実行画面。

から入力し、適当なファイル名でSAVEしてください。ここでは一応、SAVE "サーチ. N88"でSAVEしたとして説明します。

まず、LOAD "サーチ. N88" としてプログラムをロードします。ロードが終わってOKと表示がされたらRUNします。つぎにマシン語プログラムで、開始アドレスと最終アドレスの知りたいプログラムの入っているフロッピーに交換してファイル名をINPUTします。知りたいプログラム名がそのフロッピーに存在していなければFILE NOT FOUND!!!と表示されファイル名の再入力の状態に戻ります。

正しくファイル名がINPUTされていれば、直ちにそのプログラムの開始アドレスと最終アドレスを表示しますので市販されているゲームソフトなどの解析などに、利用してください。このプログラムを終わる時には、ファイル名をENDまたはendとinputすれば終わることができます。

●PセーブのUNLIST解除プログラム

次に紹介するプログラムは、Pセーブされているために LIST が見られないプログラムのリストを見たい時に便利なプログラムで

〈リスト1〉 FDDのマシン語プログラムの収容されているアドレスをサーチするプログラム

```
100 '┌──────────────────────────────────────────────┐
110 '│                                              │
120 '│  ADDRESS SEARCH PROGRAM          By.  O & K  │
130 '│                                              │
140 '│                          For   NEC PC-8801   │
150 '└──────────────────────────────────────────────┘
160 '
170 ON ERROR GOTO 490
180 DEFINT A-Z
190 WIDTH 80,25:CONSOLE 0,25,0,1:COLOR 7,0,0,7:PRINT CHR$(12)
200 COLOR 2
210 PRINT "┌─────────────────────────────────────────┐"
220 PRINT "│  ADDRESS  SEARCH        NEC PC-8801     │"
230 PRINT "│   ｵﾜﾙﾄｷﾊ FILE NAME 'END'ﾏﾀﾊ'end'ﾃﾞｽ。   │          By  O & K
240 PRINT "└─────────────────────────────────────────┘
L 1.0"
250 PRINT
260 COLOR 6:INPUT  "FDD DRIVE No.";NO
270 IF NO<1 OR NO>PEEK(&HEC7D) THEN BEEP:GOTO 260
280 PRINT
290 COLOR 7:FILES NO
300 CONSOLE 20,5,0,1 : LOCATE 0,20
310 COLOR 4:INPUT "PLEASE FILE NAME =";FILE$
320 IF FILE$="end" THEN 460
330 IF FILE$="END" THEN 460
340 GOSUB 390
350 PRINT
360 COLOR 5:PRINT "ﾌｧｲﾙ ﾈｰﾑ 「 ";FILE$;" 」 ﾉ ﾊﾞﾝﾁﾊ   ";ADD$;"   ﾊﾞﾝﾁﾆｱﾘﾏｽ。"
370 PRINT
380 GOTO 310
390 OPEN CHR$(&H30+NO)+":"+FILE$ FOR INPUT AS #1
400 AD1$=RIGHT$("000"+HEX$(CVI(INPUT$(2,1))),4)
410 AD=CVI(INPUT$(2,1))
420 AD2$=RIGHT$("000"+HEX$(AD+(AD>0)*65536!+65535!),4)
430 CLOSE #1
440 ADD$=AD1$+"-"+AD2$
450 RETURN
460 CONSOLE 0,25 : PRINT CHR$(12) : COLOR 7
470 PRINT "ｺﾞｸﾛｳｻﾏ  ｵﾜﾘ "
480 END
490 BEEP : COLOR 2
500 PRINT
510 PRINT "FILE NOT FOUND !!!"
520 PRINT
530 GOTO 310
```

◎ゲームソフトなどのマシン語エリアを知りたい時に、このプログラムを利用することでアドレスの開始番地と最終番地が表示されます。

使用方法 このプログラムをロードしRUNします。
次に目的のマシン語プログラムのファイル名を入力するとアドレスの表示がされます。このプログラムを終わるならENDと大文字または小文字で入力すると終了します。

す。市販されているゲームソフトなどはＬＩＳＴとキーボードからＩＮＰＵＴしても大半のプログラムは、無情にもピーという音と共にIllegal function callのメッセージが表示されるだけです。そこで、つぎのプログラムをキーボードからＩＮＰＵＴして間違いがなければ、ＳＡＶＥ“ＬＩＳＴＯＵＴ”としてください。これで、今までＬＩＳＴの見られなかったプログラムのリストを見ることが可能です。使い方は、次の通りです。

1．まずこのプログラムをロードします。(LOAD“LISTOUT”C／R)

2．ロードが終わってＯＫと表示が出たら次にＲＵＮします。

3．ここでＬＩＳＴの見たいプログラムをロードします。(LOAD“XXXX”)

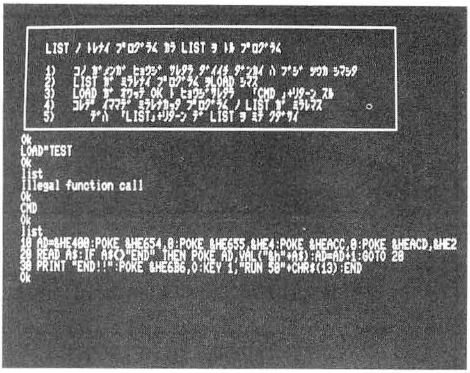
4．目的のプログラムのロードが終わってＯＫと表示が出たらキーボードより「ＣＭＤ」C/Rと入力してください。ここでまたＯＫと表示されますので、「ＬＩＳＴ」C/Rとキーボードから ＩＮＰＵＴすれば今まで見られなかったプログラムのＬＩＳＴが次々とディスプレイに表示されます。

自分で作ったプログラムをＰセーブした為にリストが見られなくなったものや、ゲームソフトなどのＢＡＳＩＣプログラムを解析する時などに便利です。ただし、一部のプログラムで動作しないものもあるかと思いますが今のところ私たちの手元にあるソフトではすべてＯＫです。短いプログラムで非常に便利なプログラムですのでぜひ、試してください。

●"パスワード"付プロテクト・プログラム

次は、自分で作ったプログラムやゲームソフトなどに、パスワードを入れないと使えなくなるようにするいじわるプログラムです。

このプログラムも少ないステップですので遊んで見てはいかがですか。？　リスト３をキーボードよりＩＮＰＵＴしてください。行番



UNLIST解除プログラムの実行画面。

号３のＢ＝１２３４５６７８９０＃となっている所が暗唱番号です。ここの所を自分の覚えやすい数字に入れ替えてください。そして、全部入れ終わったらＳＡＶＥ“ＰＳＷ”，Ａとしてアスキーセーブしておきます。「注」：Ａとしてアスキーセーブしておかないと後で他のプログラムとＭＥＲＧＥできませんので注意してください。

では、NEC PC-8801のデモソフトのQuartzを例として、このプログラムに、パスワードを付けることにします。リスト２がQuartzのリスト(PC-8801用をPC-9800漢字表示用に変更してある）です。まず先に元になるプログラムをLOADします。そしてLISTを見ます。行番号が、10より始まっていれば問題ないのですが１～９の行があるプログラム(Quartzの場合行番号が10からなので問題ありません)であればrenum10、10として行番号の変更をします。

そして、10行ごとに整理されたなら次にパスワードのプログラムと連結させます。

今、パソコンのメモリには、Quartzのプログラムがあります。これと、先程のPSWを連結するには、mergeコマンドを利用します。

では、ＰＳＷのプログラムがはいっているＦＤをドライブ１にセットしてからキーボードより、ＭＥＲＧＥ“ＰＳＷ”C/RとＩＮＰＵＴしてください。これで、ＰＳＷ＋Quartz

〈リスト2〉 LISTコマンドでIllegal function callメッセージが出てLISTの見られないプログラムを見たい人が使用するプログラム

```
100 REM  ﾄﾞｳｼﾃﾓ list ﾉ ﾐﾀｲ ｺｳｷｼﾝ ｵｳｾｲﾅ ﾋﾄﾆ ｵｸﾙ ﾌﾟﾛｸﾞﾗﾑ
110 CONSOLE 0.25.0.1 : WIDTH 80.25 : COLOR 7
120 PRINT "┌──────────────────────────────────────────────────────────────┐"
130 PRINT "│                                                              │"
140 PRINT "│   LIST ﾉ ﾄﾚﾅｲ ﾌﾟﾛｸﾞﾗﾑ ｶﾗ LIST ｦ ﾄﾙ ﾌﾟﾛｸﾞﾗﾑ                      │"
150 PRINT "│                                                              │"
160 PRINT "│   1)   ｺﾉ ｶﾞﾒﾝｶﾞ ﾋｮｳｼﾞ ｻﾚﾀﾗ ﾀﾞｲｲﾁ ﾀﾞﾝｶｲ ﾊ ﾌﾞｼﾞ ﾂｳｶ ｼﾏｼﾀ          │"
170 PRINT "│   2)   LIST ｶﾞ ﾐﾗﾚﾅｲ ﾌﾟﾛｸﾟﾗﾑ ｦ LOAD ｼﾏｽ                        │"
180 PRINT "│   3)   LOAD ｶﾞ ｵﾜｯﾃ OK ﾄ ﾋｮｳｼﾞｻﾚﾀﾗ 「 CMD 」+ﾘﾀｰﾝ ｽﾙ              │"
190 PRINT "│   4)   ｺﾚﾃﾞ ｲﾏﾏﾃﾞ ﾐﾗﾚﾅｶｯﾀ ﾌﾞﾛｸﾞﾗﾑ ﾉ LIST ｶﾞ ﾐﾗﾚﾏｽ                │"
200 PRINT "│          ﾃﾞﾊ 「LIST」+ﾘﾀｰﾝ ﾃﾞ LIST ｦ ﾐﾃ ｸﾀﾞｻｲ                  │"
210 PRINT "└──────────────────────────────────────────────────────────────┘"
220 CLEAR .&HE2FF
230 FOR I=0 TO 7
240 READ D$:D=VAL("&H"+D$):POKE &HE300+I.D
250 NEXT I
260 POKE &HEEB7.0:POKE &HEEB8.&HE3
270 DATA 3E.00.32.29.EC.C3.E5.50
```

使い方

このリストをキーボードより入力してください。

入力が終わったらまずSAVEしFDに保存します。

では手持ちのリストが見られないプログラムがあれば次の手順で操作してください。

1. このプログラムをロードする。
2. ロードが終って０Ｋが出たらLISTの見たい目的のプログラムをロードする。
3. 目的のプログラムのLOADが終わって０Ｋが表示されるまで待つ。
4. ０Ｋが表示されたらキーボードよりCMD⏎と入力する。
5. LIST⏎を入力すれば今までLISTと入力すればピー音＋Illegal function callと表示されていたプログラムがリストを表示して見られる様になる。

※ゲームソフト等の一部でリストを表示しないプログラムもありますが、大半のソフトは、このプログラムで見ることができます。

が合体しました。では、合体したプログラムをここでは、Quartzの入っていたFDにSAVEします。ファイル名をQuartzとしてsaveすれば、今までのQuartzのプログラムの内容が書き直されパスワード付きのQuartzとなります。次からQuartzをロードしても、パスワードをINPUTしないと使えなくなりました。

注意点

saveするときには、save"Quartz"、Ｐとして、saveしてください。Ｐ（大文字）を付けることでLISTを見られないように(ただし、LISTOUTプログラムを使えばバレますが…)なります。

では、ＰＳＷ＋Quartzをロードして見ましょう。ＲＵＮして、貴方が指定したパスワード以外のパスワードを一度入れてください。パスワードを投入しているときはＣＲＴはシークレット状態です。今入れたパスワードは、ＰＳＷの３行目で判定されて、Ａ＝Ｂとならないので４行目でメッセージが表示されて７行目でＮＥＷされてしまいます。では、パス

〈リスト3〉パスワード・プログラム・リスト

```
1 CONSOLE 0,25,0,0:WIDTH 80,25:COLOR 0
2 PRINT "アンショウ バンゴウ ヲ ニュウリョク シテクダサイ";
3 COLOR 1:INPUT A:B=1234567890#:IF A=B THEN 8
4 COLOR 0
5 PRINT : PRINT "アンショウ バンゴウガ チガイマス !!! "
6 PRINT "アナタハ コノプログラム ヲ シヨウデキマセン  プログラム ハ NEW サレマシタ !!!"
7 BEEP 1 : NEW
8 COLOR 0

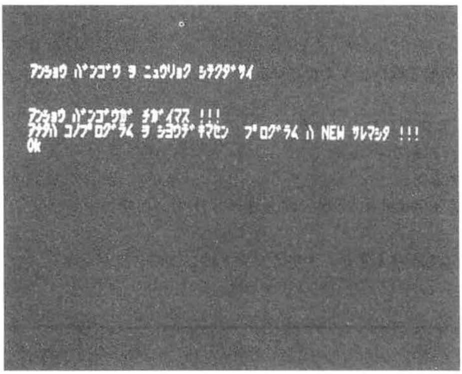


3 COLOR 1:INPUT A:B=1234567890#:IF A=B THEN 8
```

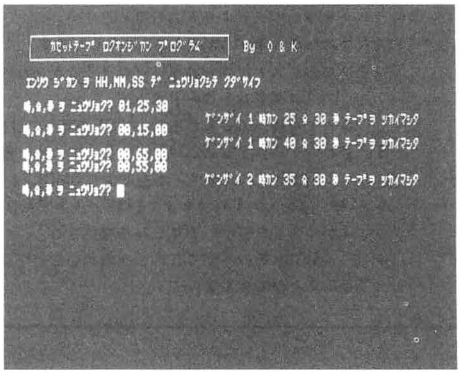
→自分のおぼえやすい数字(例)電話番号等を入れてください。

このプログラムをSAVE"PSW",A として保存してください。

→アスキーセーブ形式とする。



リスト3のプログラムを実行し、プロテクトをかけたもの。



リスト6のプログラムを実行すると…。

ワードが一致したなら3行目から8行目へ飛びCOLOR0でシークレットが解除され以後Quartzプログラムに処理を移します。

ここでは、Quartプログラムを例にお話しましたがこの要領で、他のプログラムにPSWを付けてみてはいかがですか。?

●カセットテープ用録音時間計算プログラム

最後にカセット・テープなどにレコードをダビングする時、便利なプログラムを紹介します。レコードには、録音時間が書かれているのですが、ナン分ナン秒と60進数ですから、何曲もテープにダビングするには、テープの長さと録音時間が気がかりです。

なかなか時間どおりにテープに入らずに時間が余ったり尻切れになったりした経験がありませんか。そのような時にこのプログラムで時間を計算すればうまく録音できます。〈リスト6〉

〈リスト 4〉 NEC P-8801 デモソフト Quartz を PC-9801 用に変更したソフト

```
10 '
20 ' Clock & date demonstration
30 '
40 FOR I=0 TO 7 : COLOR=(I,I) : NEXT I
50 DEFINT A-Z:DIM TIME!(2),SX!(2)
60 PI!=3.14159:PI2!=PI!*2:DA=0
70 CONSOLE 0,25,0,1:WIDTH 80,25:COLOR 7,0,0,7:SCREEN 0,3:CLS 3
80 LOCATE 25,5:PRINT "Clock & date  demonstration"
90 LOCATE 14,12:PRINT "日付 及び 時間を変更したい時は HELP ｷｰ を押して下さい。"
100 LOCATE 0,19:INPUT "hit return ",DUMY$
110 SCREEN 0,0:CLS 1
120 FOR I=0 TO 2:SX!(I)=319:SY!(I)=99:NEXT
130 ON HELP GOSUB 400:HELP STOP
140 CIRCLE(319,99),210,7,,,,4:CIRCLE(319,99),300,7,,,,33
150 TILE0$=CHR$(&HF)+CHR$(&HFF)+CHR$(&HF0)
160 TILE1$=CHR$(&H0)+CHR$(&H0)+CHR$(&HFF)
170 BGS$=STRING$(3,&H0)
180 PAINT(50,99),TILE0$,7,BGS$:PAINT(0,0),TILE1$,7,BGS$
190 PAINT(0,0),TILE1$,7,BGS$:CLS 1:GOSUB 340:SCREEN ,0:HELP ON
200 IF DA=VAL(RIGHT$(DATE$,2)) GOTO 220
210 LOCATE 36,7:PRINT DATE$;:DA=VAL(RIGHT$(DATE$,2))
220 TIME!(0)=VAL(RIGHT$(TIME$,2))+45:SW=0:CL=7:K=30:GOSUB 300
230 LOCATE 36,17:PRINT TIME$;
240 TIME!(1)=VAL(MID$(TIME$,4,2))+45:SW=1:CL=4:K=28:GOSUB 300
250 TIME!(2)=((VAL(LEFT$(TIME$,2)) MOD 12)+(TIME!(1)-45)/60)*5+45
260 SW=2:CL=3:K=20:GOSUB 300
270 CIRCLE(319,99),3,7:PAINT(319,99),7,7
280 IF TIME!(0)<>VAL(RIGHT$(TIME$,2))+45 GOTO 200 ELSE GOTO 280
290 '
300 X!=K*COS(PI2!*TIME!(SW)/60)*5+319:Y!=K*SIN(PI2!*TIME!(SW)/60)*2.5+99
310 IF (SX!(SW)<>X! OR SY!(SW)<>Y!) THEN LINE(319,99)-(SX!(SW),SY!(SW)),0
320 LINE(319,99)-(X!,Y!),CL:SX!(SW)=X!:SY!(SW)=Y!:RETURN
330 '
340 LOCATE 38,3:PRINT "１２":LOCATE 30,4:PRINT "１１":LOCATE 49,4:PRINT "１"
350 LOCATE 22,7:PRINT "１０":LOCATE 57,7:PRINT "２":LOCATE 16,12:PRINT "９"
360 LOCATE 61,12:PRINT "３":LOCATE 21,17:PRINT"８"
370 LOCATE 57,17:PRINT "４":LOCATE 29,20:PRINT"７":LOCATE 49,20:PRINT"５"
380 LOCATE 39,21:PRINT "６":RETURN
390 '
400 HELP OFF:SCREEN ,2:CLS 1
410 PRINT "OLD DATE=";DATE$:PRINT "OLD TIME=";TIME$:PRINT
420 INPUT "NEW DATE=";IN$:IN=LEN(IN$):IF IN<>0 THEN DATE$=IN$:IN$=""
430 INPUT "NEW TIME=";IN$:IN=LEN(IN$)
440 IF IN<>0 THEN TIME$=IN$:DA=VAL(RIGHT$(DATE$,2)):IN$=""
450 CLS 1:GOSUB 340:DA=0:SCREEN ,0:HELP ON:RETURN
```

※他のプログラムとPSWを連結する時の注意として桁番号に 1 ～ 9 を使用しているプログラムならRENUMで桁番号を変更してください。

(例) RENUM 10, 10 とする。

※新しいプログラムを作るなら PSW をロード後、桁番号10よりステップを進めてください。

〈リスト5〉

```
1 CONSOLE 0,25,0,0:WIDTH 80,25:COLOR 0
2 PRINT "アンショウ バンゴウ ヲ ニュウリョク シテクダサイ";
3 COLOR 1:INPUT A:B=1234567890#:IF A=B THEN 8
4 COLOR 0
5 PRINT : PRINT "アンショウ バンゴウガ チガイマス !!! "
6 PRINT "アナタハ コノプログラム ヲ シヨウデキマセン  プログラム ハ NEW サレマシタ !!!"
7 BEEP 1 : NEW
8 COLOR 0
10 '
20 ' Clock & date demonstration
30 '
40 FOR I=0 TO 7 : COLOR=(I,I) : NEXT I
50 DEFINT A-Z:DIM TIME!(2),SX!(2)
60 PI!=3.14159:PI2!=PI!*2:DA=0
70 CONSOLE 0,25,0,1:WIDTH 80,25:COLOR 7,0,0,7:SCREEN 0,3:CLS 3
80 LOCATE 25,5:PRINT "Clock & date  demonstration"
90 LOCATE 14,12:PRINT "日付  及び  時間を変更したい時は HELP キー を押して下さい。"
100 LOCATE 0,19:INPUT "hit return ",DUMY$
110 SCREEN 0,0:CLS 1
120 FOR I=0 TO 2:SX!(I)=319:SY!(I)=99:NEXT
130 ON HELP GOSUB 400:HELP STOP
140 CIRCLE(319,99),210,7,,,,4:CIRCLE(319,99),300,7,,,,33
150 TILE0$=CHR$(&HF)+CHR$(&HFF)+CHR$(&HF0)
160 TILE1$=CHR$(&H0)+CHR$(&H0)+CHR$(&HFF)
170 BGS$=STRING$(3,&H0)
180 PAINT(50,99),TILE0$,7,BGS$:PAINT(0,0),TILE1$,7,BGS$
190 PAINT(0,0),TILE1$,7,BGS$:CLS 1:GOSUB 340:SCREEN ,0:HELP ON
200 IF DA=VAL(RIGHT$(DATE$,2)) GOTO 220
210 LOCATE 36,7:PRINT DATE$;:DA=VAL(RIGHT$(DATE$,2))
220 TIME!(0)=VAL(RIGHT$(TIME$,2))+45:SW=0:CL=7:K=30:GOSUB 300
230 LOCATE 36,17:PRINT TIME$;
240 TIME!(1)=VAL(MID$(TIME$,4,2))+45:SW=1:CL=4:K=28:GOSUB 300
250 TIME!(2)=((VAL(LEFT$(TIME$,2)) MOD 12)+(TIME!(1)-45)/60)*5+45
260 SW=2:CL=3:K=20:GOSUB 300
270 CIRCLE(319,99),3,7:PAINT(319,99),7,7
280 IF TIME!(0)<>VAL(RIGHT$(TIME$,2))+45 GOTO 200 ELSE GOTO 280
290 '
300 X!=K*COS(PI2!*TIME!(SW)/60)*5+319:Y!=K*SIN(PI2!*TIME!(SW)/60)*2.5+99
310 IF (SX!(SW)<>X! OR SY!(SW)<>Y!) THEN LINE(319,99)-(SX!(SW),SY!(SW)),0
320 LINE(319,99)-(X!,Y!),CL:SX!(SW)=X!:SY!(SW)=Y!:RETURN
330 '
340 LOCATE 38,3:PRINT "１２":LOCATE 30,4:PRINT "１１":LOCATE 49,4:PRINT "１"
350 LOCATE 22,7:PRINT "１０":LOCATE 57,7:PRINT "２":LOCATE 16,12:PRINT "９"
360 LOCATE 61,12:PRINT "３":LOCATE 21,17:PRINT"８"
370 LOCATE 57,17:PRINT "４":LOCATE 29,20:PRINT"７":LOCATE 49,20:PRINT"５"
380 LOCATE 39,21:PRINT "６":RETURN
390 '
400 HELP OFF:SCREEN ,2:CLS 1
410 PRINT "OLD DATE=";DATE$:PRINT "OLD TIME=";TIME$:PRINT
420 INPUT "NEW DATE=";IN$:IN=LEN(IN$):IF IN<>0 THEN DATE$=IN$:IN$=""
430 INPUT "NEW TIME=";IN$:IN=LEN(IN$)
440 IF IN<>0 THEN TIME$=IN$:DA=VAL(RIGHT$(DATE$,2)):IN$=""
450 CLS 1:GOSUB 340:DA=0:SCREEN ,0:HELP ON:RETURN
```

NEC PC-8801 デモソフトQuartz(98)とPSWを連結し、パスワードが入力されないとRUNできなくなったソフトでフロッピーにセーブする時はSAVE "××××",P でセーブしてください。これでLISTが見られないので他人には使用できなくなります。

〈リスト6〉カセットテープ録音時間計算プログラム・リスト

```
10 CONSOLE 0,25,0,1
20 WIDTH 80,25
30 COLOR 2
40 PRINT "┌──────────────────────────────┐"
50 PRINT "│ ｶｾｯﾄﾃｰﾌﾟ ﾛｸｵﾝｼﾞｶﾝ ｹｲｻﾝ ﾌﾟﾛｸﾞﾗﾑ │  By  O & K"
60 PRINT "└──────────────────────────────┘"
70 PRINT
80 SS = 0:MM = 0:HH = 0
90 COLOR 6
100 PRINT "ｴﾝｿｳ ｼﾞｶﾝ ｦ HH,MM,SS ﾃﾞ ﾆｭｳﾘｮｸｼﾃ ｸﾀﾞｻｲ "
110 PRINT
120 COLOR 7
130 INPUT "時,分,秒 ｦ ﾆｭｳﾘｮｸ";H,M,S
140 IF S > 60  GOTO 130
150 IF M > 60  GOTO 130
160 SS = SS + S
170 IF SS > 59 GOTO  250
180 MM = MM + M
190 IF MM > 59 GOTO  280
200 HH = HH + H
210 COLOR 4
220 PRINT TAB(30);"ｹﾞﾝｻﾞｲ"   HH "時ｶﾝ" MM "分" SS "秒 ﾃｰﾌﾟｦ ﾂｶｲﾏｼﾀ"
230 GOTO 120
240 END
250 SS = SS - 60
260 MM = MM + 1
270 GOTO 180
280 MM = MM - 60
290 HH = HH + 1
300 GOTO 200
```

※レコードなどからカセットテープにダビングする時など録音時間の計算をしないとテープの途中でRECキーが上がって尻切れ録音になった方も多いことと思います。この様な時にこのプログラムで時分秒の計算をして録音をしてはいかがですか？時間を入力するたびに合計を表示しますのでテープの使用時間を知るのに便利です。

◎参考文献　PC-Techknow 8801　システムソフト……………………………㈱アスキー出版
バックアップ活用テクニック①…………………………………三才ブックス
PC8801　徹底解説…………………………………………………………ナツメ社

64K＋㊙テクニックで
ROMゲームが落とせる!!



MSX

MSXバックアッププログラム

上向井 毅

●"ロリコピーMSX"から

去年の10月に、ソフト・ハウス〝パス〞より〝ロリコピーMSX　Ver.1.0〞を発売したところ「ROMカートリッジはコピーできないのか?」「ROMカートリッジをコピーするソフトを作ってください!」などの要望があり、その期待に応えるため、半年の期間をかけROMカートリッジのバックアップ・ツール〝ロリコピーMSX Ver. 2.0/3.0〞を制作しました。

しかし、プログラムの性質上、現在までプログラム・テクニック等はマル秘にしていましたが、「MSXというコンピュータの繁栄のために」と考え、今回の発表記事となったわけです。

●ROMカートリッジのプログラムが入るスロットとアドレス

MSXでは、8ビットCPUが直接管理できる64Kバイトのメモリ空間を1個のスロットとして、基本スロットと呼ばれる4つのスロットを中心に合計16個のスロットがあります。(表1)

MSX本体にあるROMカートリッジ・スロットのほとんどは、この基本スロットです。したがって、MSX本体にセットされるROMカートリッジのプログラムは、基本スロット内にセットされているということになります。MSXのスロット・チェック・ルーチンは、電源を入れるとすぐに各基本スロットをチェックしROMカートリッジを見付けると、そのカートリッジ内のプログラムを実行します。

ROMカートリッジがセットされてない場合は、MSX BASICのインタープリタを実行するようになっています。

ここまでの説明で分かる通り、MSXは電源を入れると勝手にROMカートリッジ内のプログラムを実行してしまいます。こうなればストップもききません。

では、どうすればコピーがとれるのか?「ハードを追加すれば良い」という人がいるでしょうが、我ら無銭家にはそんなお金はありません。

そこで、MSX BASICインタープリタを実行している状態で、ROMカートリッジをセットするという方法を考えました。少し乱暴ですが、お金が無いのですから仕方がありません。その分、MSXにガンバッテもらうことにします。

この方法でROMカートリッジをセットす

表1　MSXのスロット構成

	基本スロット　00	基本スロット　01	基本スロット　02	基本スロット　03
ページ3	空　き	RAMの場合のみRAM それ以外は空き	〃	〃
C000				
ページ2	空　き	ROMカートリッジスロットの場合のみROMカートリッジのプログラムが入る。 RAMの場合はRAM（64K又は32K）	〃	〃
5000				
ページ1	MSX BASIC ROM	ROMカートリッジスロットの場合のみROMカートリッジのプログラムが入る。 RAMの場合はRAM（64Kのみ）	〃	〃
4000				
ページ0	MSX BIOS	ROMカートリッジスロットの場合のみROMカートリッジのプログラムが入る。 RAMの場合はRAM（64Kのみ）	〃	〃
0000				

	拡張スロット 1-1	拡張スロット 1-2	拡張スロット 1-3	拡張スロット 1-4	拡張スロット 2-1	拡張スロット 2-2	拡張スロット 2-3	拡張スロット 2-4	拡張スロット 3-1	拡張スロット 3-2	拡張スロット 3-3	拡張スロット 3-4	拡張スロット 4-1	拡張スロット 4-2	拡張スロット 4-3	拡張スロット 4-4
ページ3																
C000																
ページ2																
5000																
ページ1									SONY HB-701 FDのみ RAM							
4000																
ページ0																
0000																

※拡張スロットはハードの追加によって拡張しないとありません。

るのが不安な方は、バックアップMSX Ver.2に載っている〝カートリッジ・アダプタ〟を使用してください。

ＭＳＸ本体にセットするＲＯＭカートリッジのプログラムは、基本スロット内にセットされていることが分かりました。しかし、4つの基本スロットの内のどれがＲＯＭカートリッジなのでしょうか？

スロットは、ＭＳＸ ＢＡＳＩＣインタープリタが入っているので関係ありませんから、他の3つに絞られます。感の良い人は「ＲＯＭカートリッジをセットしている時とセットしていない時の状態を調べれば良い」と思うでしょう。

その通りです。こうすればＭＳＸ本体に付いているカートリッジ・スロットにどの基本スロットが、割り当てられているかを知ることができます。しかし、これだけでは情報不足です。

プログラムがどこのアドレスに入っているのかも分かりませんし、プログラムのスタート・アドレスも分かりません。しかし、ＲＯＭカートリッジのプログラムが入るアドレスは、表1の内のどれかです。

このどこに入っているかを調べるためには、プログラムの先頭アドレス（表2）から2バイトに41Hと42Hが入っているかを調べれば良いのです。また、プログラムのスタート・

アドレスは、その後の2バイトに書き込んであります。以上のことを考え合わせ、基本スロットとアドレスを調べるスロット＆アドレス・チェック・プログラムを作ってみました。(リスト2)リスト1のスロット・モニタはマシン語の分かる人がROMのプロテクトを外すツールです。

表2

先頭アドレス		終了アドレス
0000H	～	3FFFH
4000H	～	7FFFH
4000H	～	BFFFH
8000H	～	BFFFH

〈リスト1〉

```
MSX SLOT MONITOR Ver 1.0
(C)Copyright 1985 By T.Uemukai




1000 '----------------------------
1010 '-                          -
1020 '- MSX SLOT MONITOR Ver 1.0 -
1030 '-                          -
1040 '-     (C)COPYRIGHT 1985    -
1050 '-                          -
1060 '-    BY SOFT HOUSE  PASS   -
1070 '-                          -
1080 '-   PROGRAMMER  T.UEMUKAI  -
1090 '-                          -
1100 '----------------------------
1110 '
1120 '          INITIALIZE
1130 '
1140 CLEAR 200,&HDFFF
1150 SCREEN 0:COLOR 15,4,4:CLS:KEYOFF
1160 FOR I=&HE000 TO &HE036
1170 READ A$:POKE I,VAL("&H"+A$)
1180 NEXT I
1190 '
1200 ' SCREEN
1210 '
1220 PRINT"MSX SLOT MONITOR Verison 1.0"
1230 PRINT"(C)COPYRIGHT 1985 BY PASS
1240 PRINT"PROGRAMMER T.UEMUKAI"
1250 '
1260 ' COMMAND INPUT
1270 '
1280 LINE INPUT ">";CO$
1290 C1$=MID$(CO$,1,1)
1300 IF C1$="D" OR C1$="d" THEN 1340
1310 IF C1$="M" OR C1$="m" THEN 1620
1320 IF C1$="G" OR C1$="g" THEN 1770
1330 IF CO$<>"" THEN PRINT"ERR.":BEEP:PRINT:GOTO 1280 ELSE GOTO 1280
1340 '
1350 ' DUMP
1360 '
1370 D1$=MID$(CO$,2,4)
1380 D2$=MID$(CO$,7,4)
1390 D1=VAL("&H"+D1$)
1400 D2=VAL("&H"+D2$)
1410 IF D1<0 THEN D1=65536!+D1
1420 IF D2<0 THEN D2=65536!+D2
1430 FOR I=D1 TO D2 STEP 8
```

```
1440 PRINTRIGHT$("000"+HEX$(I),4);" ";
1450 FOR K=I TO I+7
1460 IF K>49151! THEN PRINT:PRINT"ADDRESS OVER.":GOTO 1280
1470 IF INKEY$=CHR$(13) THEN PRINT:PRINT"BREAK.":PRINT:GOTO 1280
1480 AD$=RIGHT$("000"+HEX$(K),4)
1490 POKE &HF000,VAL("&h"+MID$(AD$,3,2))
1500 POKE &HF001,VAL("&h"+MID$(AD$,1,2))
1510 POKE &HF002,VAL("&h"+MID$(CO$,12,2))
1520 DEFUSR=&HE000:A=USR(0)
1530 A$=HEX$(PEEK(&HF000))
1540 PRINT RIGHT$("0"+A$,2);" ";
1550 NEXT K
1560 PRINT
1570 NEXT I
1580 PRINT:GOTO 1280
1590 '
1600 ' WRITE
1610 '
1620 D1=VAL("&H"+MID$(CO$,2,4))
1630 SL=VAL("&H"+MID$(CO$,7,2))
1640 IF D1<0 THEN D1=65536!+D1
1650 FOR I=D1 TO 49151!
1660 PRINTRIGHT$("000"+HEX$(I),4);" = ";
1670 LINE INPUT DA$:DA$=MID$(DA$,1,2)
1680 IF DA$="," THEN PRINT:PRINT"BREAK.":GOTO 1280
1690 RR$=RIGHT$("000"+HEX$(I),4)
1700 POKE &HF002,VAL("&H"+DA$)
1710 POKE &HF000,VAL("&H"+MID$(RR$,3,2))
1720 POKE &HF001,VAL("&H"+MID$(RR$,1,2))
1730 POKE &HF003,SL
1740 DEFUSR=&HE020:A=USR(0)
1750 NEXT
1760 PRINT:GOTO 1280
1770 '
1780 ' JUMP
1790 '
1800 DA$=MID$(CO$,2,4):SL$=MID$(CO$,7,2)
1810 POKE &HF000,&HF7
1820 POKE &HF001,VAL("&H"+SL$)
1830 POKE &HF002,VAL("&H"+MID$(DA$,3,2))
1840 POKE &HF003,VAL("&H"+MID$(DA$,1,2))
1850 POKE &HF004,&HC9
1860 DEFUSR=&HF000:A=USR(0)
1870 END
1880 '
1890 ' MACHINE DATA
1900 '
1910 DATA E5,D5,C5,F5
1920 DATA 2A,00,F0,3A
1930 DATA 02,F0,CD,0C
1940 DATA 00,FB,32,00
1950 DATA F0,F1,C1,D1
1960 DATA E1,FB,C9,00
1970 DATA FF,FF,00,00
1980 DATA FF,FF,00,00
1990 DATA E5,D5,C5,F5
2000 DATA 2A,00,F0,3A
2010 DATA 02,F0,5F,3A
2020 DATA 03,F0,CD,14
2030 DATA 00,FB,F1,C1
2040 DATA D1,E1,C9,00
2050 '
2060 ' Program end
2070 '
```

コマンド

ダンプ：D〔先頭アドレス〕,〔終了アドレス〕,〔スロットNo.〕
書き込み：M〔書き込み先頭アドレス〕,〔スロットNo.〕
ジャンプ：G〔ジャンプ先アドレス〕,〔スロットNo.〕

〈リスト2〉

```
SLOT & ADDRESSS CHECK PROGRAM.
(C)Copyright 1985 By T.Uemukai




1000 ' SLOT & ADDR CHECK
1010 SCREEN 1:COLOR 15,4,7
1020 FOR I=&HE000 TO &HE017
1030 READ A$:POKE I,VAL("&h"+A$)
1040 NEXT
1050 PRINT"ROMｽﾀｰﾄばんち ﾁｪｯｸ ﾌﾟﾛｸﾞﾗﾑ"
1060 PRINT
1070 PRINT"(1)0000H (2)4000H (3)8000H"
1080 PRINT
1090 LINE INPUT "ｱﾄﾞﾚｽ を ｾﾚｸﾄしてください。=";A$
1100 A=VAL(A$)
1110 PRINT
1120 IF A=1 THEN A1$="00":A2$="00"
1130 IF A=2 THEN A1$="40":A2$="00"
1140 IF A=3 THEN A1$="80":A2$="00"
1150 IF A<1 OR A>3 THEN RUN
1160 FOR K=0 TO 3:' Byte
1170 FOR I=1 TO 3:' Slot
1180 POKE &HF000,VAL("&H"+A2$)+K
1190 POKE &HF001,VAL("&H"+A1$)
1200 POKE &HF002,I
1210 DEFUSR=&HE000:A=USR(0)
1220 DA(K,I)=PEEK(&HF000)
1230 NEXT I
1240 NEXT K
1250 PRINT"      ADDRESS ";A1$+A2$
1260 PRINT
1270 PRINT"SLOT   +0  +1  +2  +3"
1280 FOR I=1 TO 3
1290 PRINT I;"    ";
1300 FOR K=0 TO 3
1310 PRINTRIGHT$("0"+HEX$(DA(K,I)),2);"  ";
1320 NEXT
1330 PRINT
1340 NEXT
1350 ' ADDRESS CHECK
1360 PRINT
1370 FOR K=0 TO 3
1380 IF DA(0,K)=&H41 THEN IF DA(1,K)=&H42 THEN CH(K)=1
1390 NEXT
1400 FOR I=1 TO 3
1410 IF CH(I)=1 THEN PRINT "SLOT";I; "ROMがあります。ADDR.=";:A=DA(3,I)*&H100+DA(2,I)
:PRINTRIGHT$("000"+HEX$(A),4):AA=AA+1
1420 NEXT:PRINT
1430 IF AA=0 THEN PRINT "ROMがありません"
1440 IF AA<>0 THEN PRINT"ばんちが576FのばあいFDDがついています。"
1450 DATA E5,D5,C5,F5
1460 DATA 2A,00,F0,3A
1470 DATA 02,F0,CD,0C
1480 DATA 00,FB,32,00
1490 DATA F0,F1,C1,D1
1500 DATA E1,FB,C9,00
```

●スロット&アドレス・チェック・プログラム使用法

プログラムを入力し、タイプ・ミスが無いことを確認してテープにセーブします。MSX本体の電源を要れ、電源を入れたままROMカートリッジをセットします。もし、カートリッジ・スロット内に保護用スイッチが付いているものは、これがきかないようにしておきます。セットできたら、プログラムをロードしRUNします。

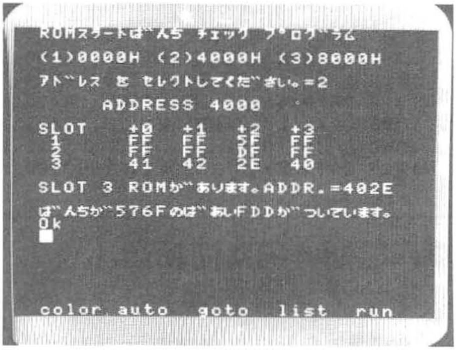
RUNしたら、3つのアドレスをセレクト

するようになっていますので、１番から順にやっています。やっているうちに、もしセレクトしたアドレスにプログラムがあれば、スロット・ナンバーとスタート・アドレスを表示します。２番（4000Ｈ）をセレクトして、ＲＯＭカートリッジがあった場合、

スロット１にＲＯＭがあります。ＡＤＤＲ＝4010

この場合、このプログラムは4000Ｈ～７ＦＦＦＨかまたは4000Ｈ～ＢＦＦＦＨに入っていて、スタート・アドレスは4010Ｈでスロットは１番ということになります。

それでは、実際にコピーしてみましょう。ＲＯＭカートリッジ内のプログラムをテープにコピーするには、ＲＯＭカートリッジ内のプログラムをＲＡＭに転送する必要があります。そこで、ＲＯＭカートリッジ内のプログラムをＲＡＭに転送するプログラムを作ってみました。（リスト３）



ROMスタート番地チェックプログラム

〈リスト３〉

```
ROM -> RAM  Program
(C)Copyright 1985 By T.Uemukai

1000 ' ROM -> RAM Program
1010 SCREEN 1:COLOR 15,4,7
1020 FOR I=&HE000 TO &HE02F
1030 READ A$:POKE I,VAL("&H"+A$)
1040 NEXT I
1050 PRINT"ROM カートリッジ から RAM にてんそうする"
1060 PRINT
1070 LINE INPUT "ROMカートリッジの はじめアドレス = ";ST$:ST=VAL("&H"+ST$)
1080 PRINT:EN$=HEX$(ST+&H3FFF)
1090 PRINT"ROMカートリッジの おわり アドレス = ";EN$:EN=VAL("&H"+EN$)
1100 PRINT
1110 LINE INPUT "ROMカートリッジの スロットばんごう= ";SL$:SL=VAL("&H"+SL$)
1120 ST$=RIGHT$("000"+HEX$(ST),4)
1130 EN$=RIGHT$("000"+HEX$(EN),4)
1140 PRINT
1150 PRINT"ST = ";ST$;" EN = ";EN$;" SL = ";SL
1160 PRINT
1170 PRINT:PRINT"てんそうちゅう..."
1180 PRINT
1190 POKE &HF000,VAL("&H"+MID$(ST$,3,2))
1200 POKE &HF001,VAL("&H"+MID$(ST$,1,2))
1210 POKE &HF002,&H0
1220 POKE &HF003,&H90
1230 POKE &HF004,&H0
1240 POKE &HF005,&H40
1250 POKE &HF006,SL
1260 DEFUSR=&HE000:A=USR(0)
1270 PRINT"9000H-CFFFH に てんそうしました。"
```

```
1280 END
1290 DATA E5,D5,C5,F5,2A,00,F0,ED
1300 DATA 5B,02,F0,ED,4B,04,F0,3A
1310 DATA 06,F0,E5,D5,C5,CD,0C,00
1320 DATA FB,C1,D1,E1,EB,77,EB,23
1330 DATA 13,0B,78,B1,28,02,18,E7
1340 DATA F1,C1,D1,E1,C9,00,00,00
```

●ROM➡RAM転送ルーチン使用法

プログラムを入力し、タイプ・ミスの無いことを確認してから、テープにセーブします。ROMカートリッジをMSXの電源を入れたままセットあるう。

セットしたら、プログラムをロードしてRUNします。

すると、ROMカートリッジのプログラム先頭アドレスのスロットを聞いてくるので、先程のプログラムで調べたデータを入力します。(最終アドレスは自動設定する)

しばらくして、BASICのコマンド待ち状態になります。この時点でプログラムはRAMに転送されています。なお、このプログラムは、9000H～CFFFHに転送するので、セーブする時は、

BSAVE "ファイル名"、&H9000、&HCFFF

とします。

注】4000H～BFFFHの場合、4000H～7FFFHと8000H～BFFFHに分けてセーブしてください。

●コピーしたプログラムのスタート法

これでコピーはできたわけですが、コピーしたプログラムをスタートさせるには、ROMカートリッジのプログラムが割り当てられていたアドレスと同じRAMアドレスに転送しなければなりません。

プログラムの内容によってアドレスが異なるために、それぞれに合ったローダーを作ってみました。(リスト4)

ここでローダーの使い方についてお話しましょう。

まず、ローダーのプログラムを入力し、テープにセーブしておきます。コピーしたROMカートリッジのプログラムと同じアドレス用のローダーをロードします。ロードしたらRUNします。

RUNしたら、裏RAMのスロット番号とプログラムのスタート・アドレスを聞いてくるので、それぞれ入力します。

注】8000H～BFFFHの場合、裏RAMスロットの入力はありません。

入力したらテープレコーダをロード状態にして、リターン・キーを押します。しばらくすると、プログラムがオート・スタートします。なお、これでもスタートしない場合はプロテクトがかかっていますから、このコピー法ではとれませんので、あしからず。

注】HB-701FDの場合は、裏スロット番号入力は"82"としてください。

〈リスト4〉

```
0000H - 3FFFH LOADER
(C)Copyright 1985 By T.Uemukai
```

```
1000 ' 0000 -> 3FFF LOADER
1010 SCREEN 1:COLOR 15,4,7
1020 PRINT"0000H - 3FFFH LOADER"
1030 FOR I=&HE000 TO &HE02F
1040 READ A$:POKE I,VAL("&H"+A$)
1050 NEXT
1060 PRINT
1070 LINE INPUT"うら RAM スロット ばんごう  =";A$
1080 PRINT
1090 LINE INPUT"プログラム の じっこうばんち =";ST$
1100 PRINT:PRINT"レコーダーをロードじょうたいにしてください。"
1110 BLOAD"CAS:"
1120 A=VAL("&H"+A$)
1130 ST$=RIGHT$("000"+ST$,4)
1140 D1$=MID$(ST$,3,2)
1150 D2$=MID$(ST$,1,2)
1160 PRINT
1170 PRINT "てんそう ちゅう..."
1180 POKE &HF000,&H0
1190 POKE &HF001,&H90
1200 POKE &HF002,&H0
1210 POKE &HF003,&H0
1220 POKE &HF004,A
1230 DEFUSR=&HE000:R=USR(0)
1240 POKE &HF000,&HF7
1250 POKE &HF001,A
1260 POKE &HF002,VAL("&H"+D1$)
1270 POKE &HF003,VAL("&H"+D2$)
1280 POKE &HF004,&HC9
1290 PRINT"じっこう..."
1300 DEFUSR=&HF000:A=USR(0)
1310 END
1320 DATA E5,D5,C5,F5,2A,00,F0,ED
1330 DATA 5B,02,F0,01,00,40,E5,D5
1340 DATA C5,7E,EB,5F,3A,04,F0,CD
1350 DATA 14,00,C1,D1,E1,23,13,0B
1360 DATA 78,B1,28,02,18,E8,F1,C1
1370 DATA D1,E1,C9,00,00,00,00,00
```

```
4000H - 7FFFH LOADER
(C)Copyright 1985 By T.Uemukai


1000 ' 4000 -> 7FFF LOADER
1010 SCREEN 1:COLOR 15,4,7
1020 PRINT"4000H - 7FFFH LOADER"
1030 FOR I=&HE000 TO &HE02F
1040 READ A$:POKE I,VAL("&H"+A$)
1050 NEXT
1060 PRINT
1070 LINE INPUT"うら RAM スロット ばんごう  =";A$
1080 PRINT
1090 LINE INPUT"プログラム の じっこうばんち =";ST$
1100 PRINT:PRINT"レコーダーをロードじょうたいにしてください。"
1110 BLOAD"CAS:"
1120 A=VAL("&H"+A$)
1130 ST$=RIGHT$("000"+ST$,4)
1140 D1$=MID$(ST$,3,2)
1150 D2$=MID$(ST$,1,2)
1160 PRINT
1170 PRINT "てんそう ちゅう..."
1180 POKE &HF000,&H0
1190 POKE &HF001,&H90
1200 POKE &HF002,&H0
1210 POKE &HF003,&H40
```

```
1220 POKE &HF004,A
1230 DEFUSR=&HE000:R=USR(0)
1240 POKE &HF000,&HF7
1250 POKE &HF001,A
1260 POKE &HF002,VAL("&H"+D1$)
1270 POKE &HF003,VAL("&H"+D2$)
1280 POKE &HF004,&HC9
1290 PRINT"じっこう..."
1300 DEFUSR=&HF000:A=USR(0)
1310 END
1320 DATA E5,D5,C5,F5,2A,00,F0,ED
1330 DATA 5B,02,F0,01,00,40,E5,D5
1340 DATA C5,7E,EB,5F,3A,04,F0,CD
1350 DATA 14,00,C1,D1,E1,23,13,0B
1360 DATA 78,B1,28,02,18,E8,F1,C1
1370 DATA D1,E1,C9,00,00,00,00,00
```

```
1000 ' 4000 -> BFFF LOADER
1010 SCREEN 1:COLOR 15,4,7
1020 PRINT"4000H - BFFFH LOADER"
1030 LOCATE 0,3:PRINTSPC(30)
1040 FOR I=&HE000 TO &HE03F
1050 READ A$:POKE I,VAL("&H"+A$)
1060 NEXT
1070 LOCATE 0,2
1080 LINE INPUT"うら RAM スロット ばんごう  =";A$
1090 LOCATE 0,4
1100 LINE INPUT"プログラム の じっこうばんち =";ST$
1110 LOCATE 0,6:PRINT "レコーダー を ロードじょうたいにしてください。"
1120 BLOAD"cas:"
1130 A=VAL("&H"+A$)
1140 ST$=RIGHT$("000"+ST$,4)
1150 D1$=MID$(ST$,3,2)
1160 D2$=MID$(ST$,1,2)
1170 LOCATE 0,8
1180 PRINT "てんそう ちゅう..."
1190 POKE &HF000,&H0
1200 POKE &HF001,&H90
1210 POKE &HF002,&H0
1220 POKE &HF003,&H40
1230 POKE &HF004,A
1240 DEFUSR=&HE000:R=USR(0)
1250 BLOAD"cas:"
1260 LOCATE 0,10:PRINTSPC(30)
1270 LOCATE 0,12:PRINT "じっこう..."
1280 POKE &HF000,&HF7:POKE &HF001,A:POKE &HF002,VAL("&h"+D1$):POKE &HF003,
VAL("&h"+D2$):POKE &HF004,&HC9
1290 DEFUSR=&HE030:A=USR(0)
1300 END
1310 DATA E5,D5,C5,F5,2A,00,F0,ED
1320 DATA 5B,02,F0,01,00,40,E5,D5
1330 DATA C5,7E,EB,5F,3A,04,F0,CD
1340 DATA 14,00,C1,D1,E1,23,13,0B
1350 DATA 78,B1,28,02,18,E8,F1,C1
1360 DATA D1,E1,C9,FF,00,00,00,00
1370 DATA 21,00,90,11,00,80,01,FF
1380 DATA 3F,ED,B0,C3,00,F0,00,00
```

```
1000 ' 8000 -> BFFF LOADER
1010 SCREEN 1:COLOR 15,4,7
1020 FOR I=&HE000 TO &HE00E
1030 READ A$:POKE I,VAL("&H"+A$)
1040 NEXT
1050 PRINT "8000H - BFFFH LOADER"
1060 PRINT
1070 LINE INPUT"フ゜ログ゛ラム し゛っこうは゛んち = ";S$
1071 PRINT:PRINT"レコータ゛ーをロート゛し゛ょうたいにしてくた゛さい。"
1072 BLOAD"CAS:"
1080 D$=RIGHT$("000"+S$,4)
1090 D1=VAL("&H"+MID$(D$,3,2))
1100 D2=VAL("&H"+MID$(D$,1,2))
1110 POKE &HE00C,D1
1120 POKE &HE00D,D2
1130 PRINT
1140 PRINT "し゛っこう..."
1150 DEFUSR=&HE000:A=USR(0)
1160 DATA 21,00,90,11,00,80,01,FF
1170 DATA 3F,ED,B0,C3,00,00,C9,00
```

●"ロリコピー"のこと

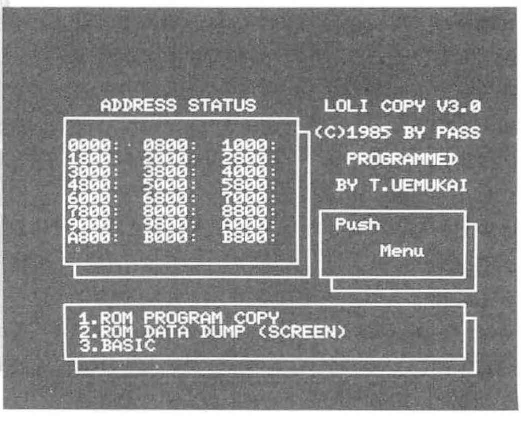
当方が発売している〝ロリコピーMSX〞は、以上の記事で紹介したテクニックをすべてサポートしています。MSXをはじめたばかりで、なにも分からない人でもワンタッチでコピーできます。定価6800円でソフト・ハウス〝パス〞より発売しています。よろしく‼

今回は64KBマシーンのみを対象としましたが、64KBでない方はRAMを64KBにしてください。

最後に、今回記事作成のためにご協力をいただきました松本無線パーツの青野さん、田原さんにお礼を申し上げます。

ソフト・ハウス〝パス〞の住所
[redacted]
[redacted] [redacted]





ロリコピーMSXと実行画面

よい子のための
マシン語入力ツール

PC-8801/mkII/SR



万能ダンプ・リスト・エディタN88

【Hexai! Editer】

櫟木　義広

●のうがき

バックアップ活用マニュアル①のテープ・バックアップ・ソフト、皆さんは入力してみましたか？（エッ本をまだ持っていない！それはまずい。すぐ買って来よう！）私も88のユーザーなので興味しんしん。さて、いっちょガンバルべと思い、〝mon〟でモニタに入りプログラムを入れ始めたのはいいがデータは間違えるし、フルキーからは打ちにくい。とさんざん苦労して入力し終え「さて、実行してみよう。フッフッフッこの私が入力ミスなどするものか！」と自己陶酔しつつ〝RUN〟「あれれ！画面が消えたゾ。オッDISKが動いた」なんと88ってやつは、機械語で暴走するとリセットが掛かるんですね！（実は今回初めて88の機械語をいじったばかりで、知らなかった）

致命傷でした。み～な消えてしもうた。ちくしょう～！APPLEはな～、APPLEは少しくらい暴走したって大丈夫なんだぞ～、くそぉ～……。

この後でチェックしてきちんと動いたのですが、まあ！おげれつ！チェックサム・プログラムの、いやN88 BASICのスピードの遅いこと。しかもチェックサムで誤り個所を発見してもらい、いちいちモニタに入り修正するという二重手間。

私は、こーゆうのはキライだ！（しかし暴走してパーになることを思えば……）しからばチェックサム付きの機械語エディタを作ってやる。もちろんBASICでは遅いのでオールマシン語だ！（なんともオールマシン語とは、いい響きだ！）

●さてさて

さて、88には機械語エディット用の〝モニタ〟というのが基本的に走るようになっています。この〝モニタ〟の機能はなかなか良くできていて（とはいっても、大したことはないが）多くのコマンドがあります。

その中に目立つのがコマンド〝E〟キーです。これは画面上のデータをカーソル・キーを使って自由に変更が効くもので、同時にASCII表示（1バイトのデータを対応するASCIIコードの文字で表すこと：別に某雑誌を宣伝しとるわけではない）もしてくれるナカナカのモノです。

大部分の人は、これを使って機械語を入力していることでしょう。しかし、残念なことにチェックサムの表示がなされていないんですね（まあ考えてみると、ユーザー側が勝手

に作ったものをメーカーがそう簡単には組まないでしょうね。まだその方式も統一されていないし……)などと考え、使い勝手を考え、扱い方はほとんど〝E〟コマンドと似せて作ってみました。

このコマンドは他の雑誌、ＡＳＣＩ、Ｉ／Ｏ、ＴＨＥ ＢＡＳＩＣ、ＯＨ／ＰＣ、Ｌｏｇｉｎなどの本のチェックサム付きリストも入力できるようになっています。

それは機械語を入力する時に、このＨｅｘａｉ／Ｅｄｉｔｅｒ（16進数のことをＨｅｘａｄｅｃｉｍａｌという。略してＨｅｘ：ヘキサと読む。へっへっへっヘキサなんてクセーぜと思いつつ作ったためヘクッサイ／エディタと名が付いたのであった)があるメモリ・エリアには当然のことながら入力ができないわけです。

従ってこのようなプログラムは、どこのアドレスに於いても動作するような（こういうのをリロケータブルなプログラムっいうっちゃ♡）形ではなくてはいけないのですが、なんせＺ80を使うのは初めてなもんで、そこまでできませんでした。スミマセーン。

ですから、異なるアドレスに書いたプログラム・リストを２種類載せておきます。それぞれ使用するメモリの位置によって使い分けるようにしてください。

リスト１のＨｅｘａｉ／ＥｄｉｔｅｒはアドレスＥ000Ｈ～Ｅ４ＦＦＨまで、リスト２ではＣ０００Ｈ～Ｃ４ＦＦＨまでを使用しますので、本書に載っているようなソフトを入力する時はリスト２の方を使い、他の雑誌の長いプログラムなどは２つに分けて入力するなどして、プログラム・エリアとＨｅｘａｉ／Ｅｄｉｔｅｒのエリアが重ならないようにしてください。

●入力しちゃおー

最初に入力する時は、仕方がないのでモニタのＥコマンドなどを使って入力してください。

①まず、88の電源を入れ、ディスクが付いていれば「Ｈｏｗ ｍａｎｙ ｆｉｌｅｓ？」ではリターン・キーを押してください。

②画面をセットします。〝ＷＩＤＴＨ 80、25：ＣＯＮＳＯＬＥ 0,25〟と打ってリターン・キーを押します。

③〝ＣＬＥＡＲ，＆ＨＣ０００〟と打ってリターン・キーを押します。

④〝ＭＯＮ〟と打ちリターンを押すとモニタに入りますので(カーソルの前の記号が〝ｈ]〟になる）リスト１を入れる時には〝ＥＥ０００〟＋リターン、リスト２なら〝ＥＣ０００〟＋リターンと打ちます。

⑤すると画面上は図１のようになりますので、フルキーボードからリスト通りに打ち込みます。この時にはキーボードの０～９、Ａ～Ｆ、カーソル、ＥＳＣ、ＳＴＯＰ、ＣＯＰＹしか働かなくなりますので、決して「びえっキーボードが壊れた」などと叫ばないように／

気を付けないといけないのは、このＥコマンドは一番下の行を入力してカーソルが右端まで行っても画面がスクロール・アップしないので同じ行に何度も書き込んでしまうことがありますから、こまめにＲＯＬＬ ＵＰ、ＲＯＬＬ ＤＯＷＮのキーを画面を動かしましょう。(ナイショ／ナイショの話。実は丹治先生もＥコマンドを使い入力していてフッと気がついたら、やっぱり一番下の行に必死に打ち込んでいたことがあるそうです)

⑥打ち込むのは図２のようにデータの部分だけです。右端と下一列の数字がチェックサムといい、横方向のデータの合計の値と縦方向のデータの合計の値とが出ており、ここの値を見るだけで、目が痛くなるような16進数の行進をながめなくて済むのですねー。

⑦入力が終わりましたら、何はともあれまずＳＡＶＥしておきましょう。ＤＩＳＫにＳＡＶＥする時は、ＣＴＲＬ＋ＢでＢＡＳＩＣに戻らないといけません。(「ＳＡＶＥのしかた」

〈リスト1〉

```
ADRS   +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [Ø123456789ABCDEF]
----------------------------------------------------------------------------------
EØØØ : CD B5 E3 DD 21 C3 E3 3E:ØØ 32 A7 E6 CD 8B 42 CD  : 6D   |ﾍｵｺﾝ!ﾃｺ> 2ｧ ﾍ B ﾍ|
EØ1Ø : C6 E2 CD B5 E3 3E ØC CD:ØD 3E CD 64 E2 CD A7 E3  : D9   |ﾆｷﾍｵｺ> ﾍ >ﾍdｷﾍｧｺ|
EØ2Ø : CD 9B E1 CD 46 E1 CD 76:E3 28 26 E6 ØF CD 2C E1  : 8Ø   |ﾍ ﾄﾍFﾄﾍvｺ(& ﾍ,ﾄ|
EØ3Ø : F5 3A D7 E3 B7 2Ø 13 F1:CB 27 CB 27 CB 27 CB 27  : 8C   |時:ﾗｺｷ 円ﾋ'ﾋ'ﾋ'ﾋ'|
EØ4Ø : F5 3E ØF A6 C1 8Ø 77 C3:A8 EØ 3E FØ A6 C1 8Ø 18  : 18   |時> ｦﾁ wﾃｨﾆ>×ｦﾁ |
EØ5Ø : F5 FE 1C CA A8 EØ FE 1D:CA C9 EØ FE 1E CA 19 E1  : CF   |時 ﾊｨﾆ ﾊﾉﾆ ﾊ ﾄ|
EØ6Ø : FE 1F CA 22 E1 Ø1 ØØ Ø1:FE F8 2Ø Ø9 2A D3 E3 Ø9  : F4   | ﾊ"ﾄ *ﾓｺ |
EØ7Ø : 22 D3 E3 18 AB FE F9 2Ø:ØA 2A D3 E3 ED 42 22 D3  : CØ   |"ﾓｺ ｫ *ﾓｺ○B"ﾓ|
EØ8Ø : E3 18 FØ FE FE 2Ø Ø3 C3:ØØ EØ FE FF 2Ø 11 CD B5  : 5D   |ｺ × ﾃ ﾆ ﾍｵ|
EØ9Ø : E3 3E Ø1 32 A7 E6 CD 9Ø:42 3E ØC CD ØD 3E FF CD  : AE   |ｺ> 2ｧ ﾍ B> ﾍ > ﾍ|
EØAØ : 2C E1 77 3E Ø1 32 D7 E3:3A D7 E3 B7 28 5E 3E ØØ  : 1E   |,ﾄw> 2ﾗｺ:ﾗｺｷ(^> |
EØBØ : 32 D7 E3 3A D5 E3 3D 21:EØ E3 BE 2Ø 56 36 ØØ 21  : 8A   |2ﾗｺ:ﾕｺ=!ｺｾ V6 !|
EØCØ : E1 E3 7E FE ØF 2Ø 4C 18:28 3A D7 E3 B7 2Ø 1A 3E  : 1E   |ｺ~ L (:ﾗｺｷ >|
EØDØ : Ø1 32 D7 E3 21 EØ E3 7E:B7 2Ø 3B 3A D5 E3 3D 77  : Ø7   | 2ﾗｺ!ｺ~ｷ ;:ﾕｺ=w|
EØEØ : 21 E1 E3 7E B7 2Ø 2F 18:15 3E ØØ 32 D7 E3 C3 2Ø  : A3   |!ｺ~ｷ / > 2ﾗｺﾃ |
EØFØ : EØ 2A D3 E3 ED 4B D5 E3:Ø9 22 D3 E3 18 FØ 2A D3  : 96   |ﾆ*ﾓｺ○Kﾕｺ "ﾓｺ ×*ﾓ|
----------------------------------------------------------------------------------
 SUM : 66 C8 96 D6 45 E7 54 5B 8E 1C Ø6 Ø6 8A A5 CC D8    FE

ADRS   +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [Ø123456789ABCDEF]
----------------------------------------------------------------------------------
E1ØØ : E3 ED 4B D5 E3 ED 42 22:D3 E3 18 E2 3E Ø1 32 D7  : 1C   |ｺ○Kﾕｺ○B"ﾓｺ ｷ> 2ﾗ|
E11Ø : E3 18 DB 34 18 D8 35 18:D5 21 E1 E3 7E B7 2Ø F6  : 4C   |ｺ ﾛ4 ﾘ5 ﾕ!ｺ~ｷ 分|
E12Ø : 18 DC 21 E1 E3 7E FE ØF:2Ø E9 18 C5 F5 2A D3 E3  : 1F   | ﾜ!ｺ~ ﾅ時*ﾓｺ|
E13Ø : 3A EØ E3 4F Ø6 ØØ Ø9 3A:E1 E3 CB 27 CB 27 CB 27  : 2F   |:ｺO :ｺﾋ'ﾋ'ﾋ'|
E14Ø : CB 27 4F Ø9 F1 C9 3E 1Ø:11 Ø8 F5 21 D8 E3 Ø1 Ø8  : 45   |ﾋ'O 円ﾉ> 時!ﾘｺ |
E15Ø : ØØ ED BØ EB Ø1 7Ø ØØ Ø9:EB 3D 2Ø EF 3A E1 E3 6F  : A6   | ○ｰ p = ＼:ｺo|
E16Ø : 26 ØØ CB 25 CB 25 CB 25:54 5D 2E ØØ 67 CB 3C CB  : ØE   |& ﾋ%ﾋ%ﾋ%T]. gﾋ<ﾋ|
E17Ø : 1D B7 ED 52 Ø1 Ø8 F5 Ø9:3A EØ E3 F5 47 CB 27 8Ø  : C5   | ｷ○R 時 :ｺ時Gﾋ' |
E18Ø : C6 Ø9 47 3A D7 E3 8Ø F5:3D 77 23 23 F1 77 F1 C6  : 98   |ﾆ G:ﾗｺ 時=w##円w円ﾆ|
E19Ø : 4Ø F5 3D 23 23 77 23 23:F1 77 C9 2A D3 E3 3E ØØ  : C4   |@時=##w##円wﾉ*ﾓｺ> |
E1AØ : 32 EB E4 FD 21 B9 F4 11:F7 F4 D5 Ø6 1Ø C5 3E ØØ  : B6   |2 !ｹ日 日ﾕ ﾅ> |
E1BØ : B7 28 Ø2 7C 85 4F 7C CD:45 E2 7D CD 45 E2 FD 23  : 32   |ｷ( ¦ O¦ﾍE }ﾍE #|
E1CØ : 3E 3A FD 77 ØØ FD 23 3A:D5 E3 47 7E F5 81 4F F1  : 79   |>: w #:ﾕｺG~時 O円|
E1DØ : 12 FD 23 CD 45 E2 13 23:1Ø F1 FD 23 3E 3A FD 77  : 69   | #ﾍE # 円 #>: w|
E1EØ : ØØ FD 23 FD 23 79 CD 45:E2 Ø1 ØØ ØØ FD Ø9 EB Ø1  : AØ   | # #yﾍE |
E1FØ : ØØ ØØ Ø9 EB C1 1Ø B6 11:78 ØØ FD 21 B8 FC E1 3A  : F1   | ﾁ ｶ x !ｸ ﾄ:|
----------------------------------------------------------------------------------
 SUM : 65 D1 97 A6 6B 73 48 73 DC EB 81 98 3D 24 B9 25    2B

ADRS   +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [Ø123456789ABCDEF]
----------------------------------------------------------------------------------
E2ØØ : D5 E3 47 C5 E5 Ø6 1Ø 3E:ØØ 86 19 1Ø FC F5 4F 3A  : 26   |ﾕｺGﾅ > 時O:|
E21Ø : EB E4 81 32 EB E4 F1 CD:45 E2 FD 23 E1 23 C1 1Ø  : 2B   | 2 円ﾍE #ﾄ#ﾁ |
E22Ø : E2 FD 23 FD 23 3A EB E4:CD 45 E2 FD 21 B3 FD CD  : BA   | # #: ﾍE !ｳ ﾍ|
E23Ø : 2C E1 7C CD 45 E2 7D CD:45 E2 FD 23 FD 23 FD 23  : 4E   |,ﾄ¦ﾍE }ﾍE # # #|
E24Ø : 7E CD 45 E2 C9 F5 CB 3F:CB 3F CB 3F CB 3F CD 58  : 7D   |~ﾍE ﾉ時ﾋ?ﾋ?ﾋ?ﾋ?ﾍX|
E25Ø : E2 F1 E6 ØF CD 58 E2 C9:32 5D E2 DD 7E ØØ FD 77  : D8   | 円 ﾍX ﾉ2] ﾝ~ w|
E26Ø : ØØ FD 23 C9 3E ØØ 32 B8:E6 CD 7A 3F CD 21 4Ø 11  : BC   | #ﾉ> 2ｸ ﾍz?ﾍ!@ |
E27Ø : C8 F3 21 E2 E3 Ø1 5Ø ØØ:ED BØ 21 41 F4 CD BD E2  : 51   |ﾈ月! P ○ｰ!A ﾍｽ |
E28Ø : 21 39 FC CD BD E2 21 D7:F4 3E 3A CD B3 E2 21 F6  : 9F   |!9 ﾍｽ !ﾗ > :ﾍｳ ! 分|
E29Ø : F4 3E 96 CD B3 E2 21 Ø7:F5 CD B3 E2 11 B2 FC 21  : 89   |> ﾍｳ ! 時ﾍｳ ｲ !|
E2AØ : 32 E4 Ø1 Ø5 ØØ ED BØ 11:A2 FD 21 37 E4 Ø1 18 ØØ  : BE   |2 ○ｰ ｢ !7 |
E2BØ : ED BØ C9 11 78 ØØ Ø6 1Ø:77 19 1Ø FC C9 Ø6 4F 3E  : FD   |○ｰﾉ x w ﾉ O>|
E2CØ : 2D 77 23 1Ø FC C9 3E ØC:CD ØD 3E 21 F2 E6 3E ØØ  : 35   |-w# ﾉ> ﾍ >! > |
E2DØ : Ø6 AØ 77 23 1Ø FC 21 4F:E4 7E B7 28 Ø6 CD ØD 3E  : 1B   | w# !O ~ｷ( ﾍ >|
E2EØ : 23 18 F6 21 F2 E6 3E FE:77 21 Ø2 E7 3C 77 CD 83  : EA   |# 分! > w! <wﾍ |
E2FØ : 35 FE FF 28 Ø6 FE FE 28:25 18 F3 3E Ø8 32 D5 E3  : E4   |5 ( (% 月> 2ﾕｺ|
----------------------------------------------------------------------------------
 SUM : B5 8B C1 89 DB AE 2B FC 76 8D 45 3F B2 12 42 F5    BC
```

```
ADRS    +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [Ø123456789ABCDEF]
------------------------------------------------------------------------------------
E3ØØ :  32 AF E1 Ø1 55 ØØ ED 43:EA E1 Ø1 7Ø ØØ ED 43 FØ  :  A4   |2ｯ ﾄ U ○C◆ﾄ p ○C×|
E31Ø :  E1 Ø1 ØØ ØØ ED 43 3A E1:ED 43 EØ E3 18 1F 3E ØØ  :  95   |ﾄ    ○C:ﾄ○C═    > |
E32Ø :  32 AF E1 3E 1Ø 32 D5 E3:Ø1 3D ØØ ED 43 EA E1 Ø1  :  34   |2ｯ ﾄ> 2ﾕｺ  = ○C◆ﾄ |
E33Ø :  68 ØØ ED 43 FØ E1 Ø1 CB:27 ED 43 3A E1 21 D4 E4  :  8Ø   |h ○C×ﾄ ﾋ'○C:ﾄ!ﾔ◢|
E34Ø :  7E B7 28 Ø6 CD ØD 3E 23:18 F6 FD 21 F6 F9 2E ØØ  :  E7   |~ｷ( ﾍ >#  ♤ !♤ . |
E35Ø :  Ø6 Ø3 CD 76 E3 28 FB E6:ØF F5 CD 58 E2 F1 B5 6F  :  58   |  ﾍvｺ( ◥ 時ﾍX±円ｵo|
E36Ø :  CB 25 CB 14 CB 25 CB 14:CB 25 CB 14 CB 25 CB 14  :  3C   |ﾋ%ﾋ ﾋ%ﾋ ﾋ%ﾋ ﾋ%ﾋ |
E37Ø :  1Ø EØ 22 D3 E3 C9 CD B5:E3 CD 83 35 F5 DB ØØ EE  :  39   | ﾆ"ﾓｺﾉﾍｵｺﾍ 5時□ ／|
E38Ø :  FF 2Ø ØA DB Ø1 EE FF 2Ø:Ø4 F1 BF 18 1A F1 FE 2C  :  13   |   □ ／   円ｿ  円 ,|
E39Ø :  2Ø Ø5 3E ØF B7 18 1Ø FE:2E 2Ø Ø5 B7 3E ØE 18 Ø7  :  C4   |  > ｷ    .  ｷ>    |
E3AØ :  FE 3D 2Ø FA B7 3E ØC F3:F5 3A C2 E6 F6 Ø6 D3 31  :  2Ø   | =   ｷ> 月時:ﾂ◥♤ ﾓ1|
E3BØ :  32 C2 E6 F1 C9 F5 3A C2:E6 E6 F9 D3 31 32 C2 E6  :  28   |2ﾂ◥円ﾉ時:ﾂ◥◥ ﾓ12ﾂ◥|
E3CØ :  F1 FB C9 3Ø 31 32 33 34:35 36 37 38 39 41 42 43  :  88   |円 ﾉØ123456789ABC|
E3DØ :  44 45 46 ØØ ØØ ØØ ØØ ØØ:8Ø ØØ ØØ Ø6 ØØ ØØ ØØ Ø6  :  5B   |DEF             |
E3EØ :  ØØ ØØ 2Ø 41 44 52 53 2Ø:2Ø 2Ø 2B 3Ø 2Ø 2B 31 2Ø  :  A1   |   ADRS    +Ø +1 |
E3FØ :  2B 32 2Ø 2B 33 2Ø 2B 34:2Ø 2B 35 2Ø 2B 36 2Ø 2B  :  A6   |+2 +3 +4 +5 +6 +|
------------------------------------------------------------------------------------
 SUM :  BB B4 2E 56 8Ø 56 D4 FF D6 DD 52 52 D7 DA 22 24     EA

ADRS    +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [Ø123456789ABCDEF]
------------------------------------------------------------------------------------
E4ØØ :  37 2Ø 2B 38 2Ø 2B 39 2Ø:2B 41 2Ø 2B 42 2Ø 2B 43  :  E5   |7 +8 +9 +A +B +C|
E41Ø :  2Ø 2B 44 2Ø 2B 45 2Ø 2B:46 2Ø 2Ø 53 55 4D 2Ø 2Ø  :  25   | +D +E +F  SUM  |
E42Ø :  5B 3Ø 31 32 33 34 35 36:37 38 39 41 42 43 44 45  :  B7   |[Ø123456789ABCDE|
E43Ø :  46 5D 53 55 4D 2Ø 3A 44:61 74 61 2Ø 41 64 64 72  :  Ø7   |F]SUM :Data Addr|
E44Ø :  65 73 73 2Ø 3D 2Ø 26 68:2Ø 2Ø 2Ø 2Ø 2Ø 3A 2Ø 2A  :  7A   |ess = &h     : *|
E45Ø :  2A 2Ø 48 45 58 41 49 2Ø:21 2Ø 45 64 69 74 65 72  :  77   |* HEXAI ! Editer|
E46Ø :  2Ø 2A 2A 2Ø 2Ø 2Ø 42 79:2Ø 59 6F 73 68 69 2E 49  :  32   | **   By Yoshi.I|
E47Ø :  63 68 69 6B 69 ØD ØA 1F:1F 2Ø 2Ø 4D 6F 64 65 2Ø  :  42   |chiki     Mode |
E48Ø :  53 65 6C 65 63 74 ØD ØA:1F 2Ø 66 2E 31 2Ø 2D 2Ø  :  E8   |Select    f.1 - |
E49Ø :  52 61 64 69 6F 2Ø 4C 69:66 65 2C 49 2F 4F 2C 54  :  Ø2   |Radio Life,I/O,T|
E4AØ :  48 45 2Ø 42 41 53 49 43:2C 65 74 63 2E 2E ØD ØA  :  EA   |HE BASIC,etc..  |
E4BØ :  1F 2Ø 66 2E 32 2Ø 2D 2Ø:41 53 43 49 49 ØD ØA 1F  :  11   |  f.2 - ASCII   |
E4CØ :  1F 2A 2Ø 5Ø 72 65 73 73:2Ø 4B 65 79 2E 2E 2E ØD  :  56   | * Press Key... |
E4DØ :  ØA 1F 1F ØØ 3E 3E 3E 2Ø:45 6E 74 72 79 2Ø 41 64  :  F9   |     >>> Entry Ad|
E4EØ :  64 72 65 73 73 2Ø 3D 2Ø:26 68 ØØ ØØ 42 79 2Ø 59  :  6Ø   |dress = &h  By Y|
E4FØ :  2E 49 63 68 69 6B 69 2Ø:3Ø 38 2F 33 31 2F 38 35  :  36   |.Ichiki Ø8/31/85|
------------------------------------------------------------------------------------
 SUM :  D1 2C 9E 38 BA 87 A9 8E 36 5C 1F 64 6B 2F 42 BB     F7
```

〈図１〉モニタのEコマンドの画面

```
B100 C3 20 B0 2A BD B3 ED 4B BF B3 ED 42 22 BD B3 C3    ﾃ -*ｽｳCKｿｳCB"ｽｳﾃ
B110 20 B0 3E 01 32 C1 B3 C3 20 B0 34 C3 20 B0 35 C3     -> 2ﾁｳﾃ -4ﾃ -5ﾃ
B120 20 B0 21 CB B3 7E B7 20 F5 C3 03 B1 21 CB B3 7E     -!ﾋｳ~ｷ 時ﾃ ｱ!ﾋｳ~
B130 FE 0F 20 E6 C3 F5 B0 F5 2A BD B3 3A CA B3 4F 06      ◥ﾃ時-時*ｽｳ:ﾊｳO
B140 00 09 3A CB B3 00 00 CB 27 CB 27 CB 27 4F 09 F1      :ﾋｳ  ﾋ'ﾋ'ﾋ'O 円
B150 C9 3E 10 11 08 F5 21 C2 B3 01 08 00 ED B0 EB 01    ﾉ>    時!ﾂｳ    ○-♠
B160 70 00 09 EB 3D 20 EF 3A CB B3 6F 26 00 CB 25 CB    p  ♠= ＼:ﾋｳo& ﾋ%ﾋ
B170 25 CB 25 54 5D 2E 00 67 CB 3C CB 1D B7 ED 52 01    %ﾋ%T]. gﾋ<ﾋ ｷ○R
B180 08 F5 09 3A CA B3 F5 47 CB 27 80 C6 09 47 3A C1     時 :ﾊｳ時Gﾋ'_ﾆ G:ﾁ
B190 B3 80 F5 3D 77 23 23 F1 77 F1 C6 40 F5 3D 23 23    ｳ_時=w##円w円ﾆ@時=##
B1A0 77 23 23 F1 77 C9 2A BD B3 3E 00 32 D5 B4 FD 21    w##円wﾉ*ｽｳ> 2ﾕｴ !
B1B0 B9 F4 11 F7 F4 D5 06 10 C5 3E 08 B7 28 02 7C 85    ｹ日 ♠日ﾕ  ﾅ> ｷ( |■
B1C0 4F 7C CD 50 B2 7D CD 50 B2 FD 23 3E 3A FD 77 00    O|ﾍPｲ}ﾍPｲ #>: w
B1D0 FD 23 3A BF B3 47 7E F5 81 4F F1 12 FD 23 CD 50     #:ｿｳG~時_O円  #ﾍP
B1E0 B2 13 23 10 F1 FD 23 3E 3A FD 77 00 FD 23 FD 23    ｲ # 円 #>: w   # #
B1F0 79 CD 50 B2 01 55 00 FD 09 EB 01 70 00 09 EB C1    yﾍPｲ U    ♠ p   ♠ﾁ
```

〈リスト 2〉

```
ADRS   +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM  [Ø123456789ABCDEF]
------------------------------------------------------------------------------------
CØØØ : CD B5 C3 DD 21 C3 C3 3E:ØØ 32 A7 E6 CD 8B 42 CD : 2D  |ヘオテン!テテ> 2ァ◥ヘ B'ヘ|
CØ1Ø : C6 C2 CD B5 C3 3E ØC CD:ØD 3E CD 64 C2 CD A7 C3 : 59  |ニツヘオテ> ヘ >ヘdツヘァテ|
CØ2Ø : CD 9B C1 CD 46 C1 CD 76:C3 28 26 E6 ØF CD 2C C1 : ØØ  |ヘ チヘFチヘvテ(&◥ ヘ,チ|
CØ3Ø : F5 3A D7 C3 B7 2Ø 13 F1:CB 27 CB 27 CB 27 CB 27 : 6C  |畤:ラテキ  円ヒ'ヒ'ヒ'ヒ'|
CØ4Ø : F5 3E ØF A6 C1 8Ø 77 C3:A8 CØ 3E FØ A6 C1 8Ø 18 : F8  |畤> ヲチ wティタ>×ヲチ  |
CØ5Ø : F5 FE 1C CA A8 CØ FE 1D:CA C9 CØ FE 1E CA 19 C1 : 6F  |畤  ハィタ  ハノタ  ハ チ|
CØ6Ø : FE 1F CA 22 C1 Ø1 ØØ Ø1:FE F8 2Ø Ø9 2A D3 C3 Ø9 : B4  |  ハ"チ        *モテ |
CØ7Ø : 22 D3 C3 18 AB FE F9 2Ø:ØA 2A D3 C3 ED 42 22 D3 : 8Ø  |"モテ ォ     *モテ□B"モ|
CØ8Ø : C3 18 FØ FE FE 2Ø Ø3 C3:ØØ CØ FE FF 2Ø 11 CD B5 : 1D  |テ ×     テ タ    ヘオ|
CØ9Ø : C3 3E Ø1 32 A7 E6 CD 9Ø:42 3E ØC CD ØD 3E FF CD : 8E  |テ> 2ァ◥ヘ B> ヘ > ヘ|
CØAØ : 2C C1 77 3E Ø1 32 D7 C3:3A D7 C3 B7 28 5E 3E ØØ : BE  |,チw> 2ラテ:ラテキ(^> |
CØBØ : 32 D7 C3 3A D5 C3 3D 21:EØ C3 BE 2Ø 56 36 ØØ 21 : 2A  |2ラテ:ユテ=!ニテセ V6 !|
CØCØ : E1 C3 7E FE ØF 2Ø 4C 18:28 3A D7 C3 B7 2Ø 1A 3E : DE  |ヒテ~   L (:ラテキ  >|
CØDØ : Ø1 32 D7 C3 21 EØ C3 7E:B7 2Ø 3B 3A D5 C3 3D 77 : A7  | 2ラテ!ニテ~キ ;:ユテ=w|
CØEØ : 21 E1 C3 7E B7 2Ø 2F 18:15 3E ØØ 32 D7 C3 C3 2Ø : 63  |!ヒテ~キ /  > 2ラテテ |
CØFØ : CØ 2A D3 C3 ED 4B D5 C3:Ø9 22 D3 C3 18 FØ 2A D3 : 16  |タ*モテ□Kユテ "モテ ×*モ|
------------------------------------------------------------------------------------
 SUM : Ø6 68 F6 76 Ø5 87 14 1B 6E BC C6 A6 6A 65 AC 78   1E

ADRS   +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM  [Ø123456789ABCDEF]
------------------------------------------------------------------------------------
C1ØØ : C3 ED 4B D5 C3 ED 42 22:D3 C3 18 E2 3E Ø1 32 D7 : BC  |テ□Kユテ□B"モテ ヰ> 2ラ|
C11Ø : C3 18 DB 34 18 D8 35 18:D5 21 E1 C3 7E B7 2Ø F6 : ØC  |テ ロ4 リ5 ユ!ヒテ~キ ♠|
C12Ø : 18 DC 21 E1 C3 7E FE ØF:2Ø E9 18 C5 F5 2A D3 C3 : DF  | ワ!ヒテ~    ♥ ナ畤*モテ|
C13Ø : 3A EØ C3 4F Ø6 ØØ Ø9 3A:E1 C3 CB 27 CB 27 CB 27 : EF  |:ニテO    :ヒテヒ'ヒ'ヒ'|
C14Ø : CB 27 4F Ø9 F1 C9 3E 1Ø:11 Ø8 F5 21 D8 C3 Ø1 Ø8 : 25  |ヒ'O 円ノ>   畤!リテ  |
C15Ø : ØØ ED BØ EB Ø1 7Ø ØØ Ø9:EB 3D 2Ø EF 3A E1 C3 6F : 86  | □ー♣ p  ♣= \:ヒテo|
C16Ø : 26 ØØ CB 25 CB 25 CB 25:54 5D 2E ØØ 67 CB 3C CB : ØE  |& ヒ%ヒ%ヒ%T]. gヒ<ヒ|
C17Ø : 1D B7 ED 52 Ø1 Ø8 F5 Ø9:3A EØ C3 F5 47 CB 27 8Ø : A5  | キ□R  畤 :ニテ畤Gヒ' |
C18Ø : C6 Ø9 47 3A D7 C3 8Ø F5:3D 77 23 23 F1 77 F1 C6 : 78  |ニ G:ラテ 畤=w##円w円ニ|
C19Ø : 4Ø F5 3D 23 23 77 23 23:F1 77 C9 2A D3 C3 3E ØØ : A4  |@畤=##w##円wノ*モテ> |
C1AØ : 32 EB C4 FD 21 B9 F4 11:F7 F4 D5 Ø6 1Ø C5 3E ØØ : 96  |2♣ト !ケ日 ♣日ユ  ナ> |
C1BØ : B7 28 Ø2 7C 85 4F 7C CD:45 C2 7D CD 45 C2 FD 23 : F2  |キ( ¦ O¦ヘEツ}ヘEツ #|
C1CØ : 3E 3A FD 77 ØØ FD 23 3A:D5 C3 47 7E F5 81 4F F1 : 59  |>: w  #:ユテG~畤 O円|
C1DØ : 12 FD 23 CD 45 C2 13 23:1Ø F1 FD 23 3E 3A FD 77 : 49  | #ヘEツ # 円 #>: w|
C1EØ : ØØ FD 23 FD 23 79 CD 45:C2 Ø1 ØØ ØØ FD Ø9 EB Ø1 : 8Ø  | # #yヘEツ       ♣ |
C1FØ : ØØ ØØ Ø9 EB C1 1Ø B6 11:78 ØØ FD 21 B8 FC E1 3A : F1  |   ♣チ カ x  !ク ヒ:|
------------------------------------------------------------------------------------
 SUM : 25 D1 57 A6 2B 33 48 73 BC 6B 61 78 3D C4 99 Ø5   AB

ADRS   +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM  [Ø123456789ABCDEF]
------------------------------------------------------------------------------------
C2ØØ : D5 C3 47 C5 E5 Ø6 1Ø 3E:ØØ 86 19 1Ø FC F5 4F 3A : Ø6  |ユテGナ◣  >       畤O:|
C21Ø : EB C4 81 32 EB C4 F1 CD:45 C2 FD 23 E1 23 C1 1Ø : CB  |♣ト 2♣ト円ヘEツ #ヒ#チ |
C22Ø : E2 FD 23 FD 23 3A EB C4:CD 45 C2 FD 21 B3 FD CD : 7A  |ヰ # #:♣トヘEツ !ウ ヘ|
C23Ø : 2C C1 7C CD 45 C2 7D CD:45 C2 FD 23 FD 23 FD 23 : EE  |,チ¦ヘEツ}ヘEツ # # #|
C24Ø : 7E CD 45 C2 C9 F5 CB 3F:CB 3F CB 3F CB 3F CD 58 : 5D  |~ヘEツノ畤ヒ?ヒ?ヒ?ヒ?ヘX|
C25Ø : C2 F1 E6 ØF CD 58 C2 C9:32 5D C2 DD 7E ØØ FD 77 : 78  |ツ円◥ ヘXツノ2]ツン~  w|
C26Ø : ØØ FD 23 C9 3E ØØ 32 B8:E6 CD 7A 3F CD 21 4Ø 11 : BC  | #ノ> 2ク◥ヘz?ヘ!@ |
C27Ø : C8 F3 21 E2 C3 Ø1 5Ø ØØ:ED BØ 21 41 F4 CD BD C2 : 11  |ネ月!ヰテ P □ー!A日ヘスツ|
C28Ø : 21 39 FC CD BD C2 21 D7:F4 3E 3A CD B3 C2 21 F6 : 5F  |!9 ヘスツ!ラ日>:ヘウツ!♠|
C29Ø : F4 3E 96 CD B3 C2 21 Ø7:F5 CD B3 C2 11 B2 FC 21 : 49  |日> ヘウツ! 畤ヘウツ イ !|
C2AØ : 32 C4 Ø1 Ø5 ØØ ED BØ 11:A2 FD 21 37 C4 Ø1 18 ØØ : 7E  |2ト   □ー 「 !7ト   |
C2BØ : ED BØ C9 11 78 ØØ Ø6 1Ø:77 19 1Ø FC C9 Ø6 4F 3E : FD  |□ーノ x   w   ノ O>|
C2CØ : 2D 77 23 1Ø FC C9 3E ØC:CD ØD 3E 21 F2 E6 3E ØØ : 35  |-w#  ノ> ヘ >!年◥> |
C2DØ : Ø6 AØ 77 23 1Ø FC 21 4F:C4 7E B7 28 Ø6 CD ØD 3E : FB  |  w#  !Oト~キ( ヘ >|
C2EØ : 23 18 F6 21 F2 E6 3E FE:77 21 Ø2 E7 3C 77 CD 83 : EA  |# ♠!年◥> w! ◤<wヘ |
C2FØ : 35 FE FF 28 Ø6 FE FE 28:25 18 F3 3E Ø8 32 D5 C3 : C4  |5  (    (% 月> 2ユテ|
------------------------------------------------------------------------------------
 SUM : 95 ØB C1 69 BB 2E ØB DC 56 4D Ø5 1F 92 F2 42 B5   DC
```

```
ADRS    +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [Ø123456789ABCDEF]
-----------------------------------------------------------------------------------
C3ØØ  : 32 AF C1 Ø1 55 ØØ ED 43:EA C1 Ø1 7Ø ØØ ED 43 FØ  :  64    |2ｯﾁ U ○C◆ﾁ p ○C×|
C31Ø  : C1 Ø1 ØØ ØØ ED 43 3A C1:ED 43 EØ C3 18 1F 3E ØØ  :  35    |ﾁ   ○C:ﾁ○C二ﾃ   > |
C32Ø  : 32 AF C1 3E 1Ø 32 D5 C3:Ø1 3D ØØ ED 43 EA C1 Ø1  :  D4    |2ｯﾁ> 2ﾕﾃ = ○C◆ﾁ  |
C33Ø  : 68 ØØ ED 43 FØ C1 Ø1 CB:27 ED 43 3A C1 21 D4 C4  :  2Ø    |h ○C×ﾁ ﾋ'○C:ﾁ!ﾔﾄ|
C34Ø  : 7E B7 28 Ø6 CD ØD 3E 23:18 F6 FD 21 F6 F9 2E ØØ  :  E7    |~ｷ( ﾍ >#  !  .  |
C35Ø  : Ø6 Ø3 CD 76 C3 28 FB E6:ØF F5 CD 58 C2 F1 B5 6F  :  18    |  ﾍvﾃ(  ﾍX ﾂ ｵo|
C36Ø  : CB 25 CB 14 CB 25 CB 14:CB 25 CB 14 CB 25 CB 14  :  3C    |ﾋ%ﾋ ﾋ%ﾋ ﾋ%ﾋ ﾋ%ﾋ |
C37Ø  : 1Ø EØ 22 D3 C3 C9 CD B5:C3 CD 83 35 F5 DB ØØ EE  :  F9    | ﾆ"ﾓﾃﾉﾍｵﾃﾍ 5 ﾛ  |
C38Ø  : FF 2Ø ØA DB Ø1 EE FF 2Ø:Ø4 F1 BF 18 1A F1 FE 2C  :  13    |   ﾛ       ,|
C39Ø  : 2Ø Ø5 3E ØF B7 18 1Ø FE:2E 2Ø Ø5 B7 3E ØE 18 Ø7  :  C4    |  > ｷ   .  ｷ>    |
C3AØ  : FE 3D 2Ø FA B7 3E ØC F3:F5 3A C2 E6 F6 Ø6 D3 31  :  2Ø    | =  ｷ>  :ﾂ   ﾓ1|
C3BØ  : 32 C2 E6 F1 C9 F5 3A C2:E6 E6 F9 D3 31 32 C2 E6  :  28    |2ﾂ ﾉ :ﾂ   ﾓ12ﾂ |
C3CØ  : F1 FB C9 3Ø 31 32 33 34:35 36 37 38 39 41 42 43  :  88    | ﾉØ123456789ABC|
C3DØ  : 44 45 46 ØØ ØØ ØØ ØØ ØØ:8Ø ØØ ØØ Ø6 ØØ ØØ ØØ Ø6  :  5B    |DEF             |
C3EØ  : ØØ ØØ 2Ø 41 44 52 53 2Ø:2Ø 2Ø 2B 3Ø 2Ø 2B 31 2Ø  :  A1    |   ADRS    +Ø +1 |
C3FØ  : 2B 32 2Ø 2B 33 2Ø 2B 34:2Ø 2B 35 2Ø 2B 36 2Ø 2B  :  A6    |+2 +3 +4 +5 +6 +|
-----------------------------------------------------------------------------------
 SUM  : 9B B4 EE 56 4Ø 36 D4 BF B6 BD 52 32 97 DA Ø2 Ø4     ØA

ADRS    +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM   [Ø123456789ABCDEF]
-----------------------------------------------------------------------------------
C4ØØ  : 37 2Ø 2B 38 2Ø 2B 39 2Ø:2B 41 2Ø 2B 42 2Ø 2B 43  :  E5    |7 +8 +9 +A +B +C|
C41Ø  : 2Ø 2B 44 2Ø 2B 45 2Ø 2B:46 2Ø 2Ø 53 55 4D 2Ø 2Ø  :  25    | +D +E +F  SUM  |
C42Ø  : 5B 3Ø 31 32 33 34 35 36:37 38 39 41 42 43 44 45  :  B7    |[Ø123456789ABCDE|
C43Ø  : 46 5D 53 55 4D 2Ø 3A 44:61 74 61 2Ø 41 64 64 72  :  Ø7    |F]SUM :Data Addr|
C44Ø  : 65 73 73 2Ø 3D 2Ø 26 68:2Ø 2Ø 2Ø 2Ø 2Ø 3A 2Ø 2A  :  7A    |ess = &h     : *|
C45Ø  : 2A 2Ø 48 45 58 41 49 2Ø:21 2Ø 45 64 69 74 65 72  :  77    |* HEXAI ! Editer|
C46Ø  : 2Ø 2A 2A 2Ø 2Ø 2Ø 42 79:2Ø 59 6F 73 68 69 2E 49  :  32    | **   By Yoshi.I|
C47Ø  : 63 68 69 6B 69 ØD ØA 1F:1F 2Ø 2Ø 4D 6F 64 65 2Ø  :  42    |chiki     Mode  |
C48Ø  : 53 65 6C 65 63 74 ØD ØA:1F 2Ø 66 2E 31 2Ø 2D 2Ø  :  E8    |Select    f.1 - |
C49Ø  : 52 61 64 69 6F 2Ø 4C 69:66 65 2C 49 2F 4F 2C 54  :  Ø2    |Radio Life,I/O,T|
C4AØ  : 48 45 2Ø 42 41 53 49 43:2C 65 74 63 2E 2E ØD ØA  :  EA    |HE BASIC,etc..  |
C4BØ  : 1F 2Ø 66 2E 32 2Ø 2D 2Ø:41 53 43 49 49 ØD ØA 1F  :  11    |  f.2 - ASCII    |
C4CØ  : 1F 2A 2Ø 5Ø 72 65 73 73:2Ø 4B 65 79 2E 2E 2E ØD  :  56    | * Press Key...  |
C4DØ  : ØA 1F 1F ØØ 3E 3E 3E 2Ø:45 6E 74 72 79 2Ø 41 64  :  F9    |     >>> Entry Ad|
C4EØ  : 64 72 65 73 73 2Ø 3D 2Ø:26 68 ØØ ØØ 42 79 2Ø 59  :  6Ø    |dress = &h  By Y|
C4FØ  : 2E 49 63 68 69 6B 69 2Ø:3Ø 38 2F 33 31 2F 38 35  :  36    |.Ichiki Ø8/31/85|
-----------------------------------------------------------------------------------
 SUM  : D1 2C 9E 38 BA 87 A9 8E 36 5C 1F 64 6B 2F 42 BB     F7
```

〈図 2〉 入力の説明

```
                         ここだけを入力する
ADRS    +Ø +1 +2 +3 +4 +5 +6 +7|+8 +9 +A +B +C +D +E +F   SUM   [Ø123456789ABCDEF]
-------------------------------↓---------------------------------------------------
CØØØ  : CD B5 C3 DD 21 C3 C3 3E:ØØ 32 A7 E6 CD 8B 42 CD  :  2D    |ﾍｵﾃﾝ!ﾃﾃ> 2ｧ ﾍ Bﾍ|
CØ1Ø  : C6 C2 CD B5 C3 3E ØC CD:ØD 3E CD 64 C2 CD A7 C3  :  59    |ﾆﾂﾍｵﾃ> ﾍ >ﾍdﾂﾍｧﾃ|
CØ2Ø  : CD 9B C1 CD 46 C1 CD 76:C3 28 26 E6 ØF CD 2C C1  :  ØØ    |ﾍ ﾁﾍFﾁﾍvﾃ(& ﾍ,ﾁ|
CØ3Ø  : F5 3A D7 C3 B7 2Ø 13 F1:CB 27 CB 27 CB 27 CB 27  :  6C    | :ﾗﾃｷ   ﾋ'ﾋ'ﾋ'ﾋ'|
CØ4Ø  : F5 3E ØF A6 C1 8Ø 77 C3:A8 CØ 3E FØ A6 C1 8Ø 18  :  F8    | > ｦﾁ wﾃｨﾀ> ｦﾁ  |
CØ5Ø  : F5 FE 1C CA A8 CØ FE 1D:CA C9 CØ FE 1E CA 19 C1  :  6F    |   ﾊｨﾀ  ﾊﾉﾀ  ﾊ ﾁ|
CØ6Ø  : FE 1F CA 22 C1 Ø1 ØØ Ø1:FE F8 2Ø Ø9 2A D3 C3 Ø9  :  B4    |  ﾊ"ﾁ       *ﾓﾃ |
CØ7Ø  : 22 D3 C3 18 AB FE F9 2Ø:ØA 2A D3 C3 ED 42 22 D3  :  8Ø    |"ﾓﾃ ｫ     *ﾓﾃ B"ﾓ|
CØ8Ø  : C3 18 FØ FE FE 2Ø Ø3 C3:ØØ CØ FE FF 2Ø 11 CD B5  :  1D    |ﾃ       ﾃ ﾀ    ﾍｵ|
CØ9Ø  : C3 3E Ø1 32 A7 E6 CD 9Ø:42 3E ØC CD ØD 3E FF CD  :  8E    |ﾃ> 2ｧ ﾍ B> ﾍ > ﾍ|
CØAØ  : 2C C1 77 3E Ø1 32 D7 C3:3A D7 C3 B7 28 5E 3E ØØ  :  BE    |,ﾁw> 2ﾗﾃ:ﾗﾃｷ(^> |
CØBØ  : 32 D7 C3 3A D5 C3 3D 21:EØ C3 BE 2Ø 56 36 ØØ 21  :  2A    |2ﾗﾃ:ﾕﾃ=! ﾃｾ V6 !|
CØCØ  : E1 C3 7E FE ØF 2Ø 4C 18:28 3A D7 C3 B7 2Ø 1A 3E  :  DE    | ﾃ~   L (:ﾗﾃｷ  >|
CØDØ  : Ø1 32 D7 C3 21 EØ C3 7E:B7 2Ø 3B 3A D5 C3 3D 77  :  A7    | 2ﾗﾃ! ﾃ~ｷ ;:ﾕﾃ=w|
CØEØ  : 21 E1 C3 7E B7 2Ø 2F 18:15 3E ØØ 32 D7 C3 C3 2Ø  :  63    |! ﾃ~ｷ /  > 2ﾗﾃﾃ |
CØFØ  : CØ 2A D3 C3 ED 4B D5 C3:Ø9 22 D3 C3 18 FØ 2A D3  :  16    |ﾀ*ﾓﾃ Kﾕﾃ "ﾓﾃ  *ﾓ|
-----------------------------------------------------------------------------------
 SUM  : Ø6 68 F6 76 Ø5 87 14 1B 6E BC C6 A6 6A 65 AC 78     1E  ← この部分がチェックサムで、
                                                                  これは入力しない。
 Data Address = &HCØØØ : CD
```

〈図３〉Hexai！ Editerの画面

```
** HEXAI ! Editer **   By Yoshi.Ichiki

 Mode Select

 f.1 - Radio Life,I/O,THE BASIC,etc...

 f.2 - ASCII

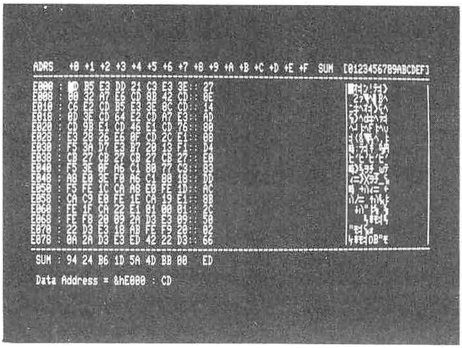
* Press Key...

>>> Entry Address = &H
```

を見てください）暴走族にキミの努力の結晶を奪われてもいいなら私は知りませんよ！

さらにバックアップ活用テクニック①のＰ.22に載っているチェックサム・プログラムを走らせます。

Start Address？にはＥ０００と打ち込みＥ５００Ｈまでチェックサムが合っているか確かめ、もし違っているところがあったならＳＴＯＰを押して再び④⑤⑦の手順で入力、修正を繰り返します。



このように画面表示する。

⑧すべて正確に入力したなら、もう一度ＳＡＶＥしておきましょう。そうしたらモニタに戻して〝ＧＥ０００〞＋リターンです。（リスト１の時。リスト２を入力したのならＧＣ０００＋リターン）

うまく動作すれば図３のような表示になるはずです。図３に従って操作してください。

希望アドレスの上位３ケタを入力するとすぐに図４の画面になります。入力方法は、図５の通り、通常の16進データはテンキーを使い文字のＡＳＣＩＩコードを書き込む時はフルキーから書きたいキーを押すと、そのアドレスに直接書き込まれます。

その辺のフィーリングは実際に操作してつかんでみてください。その際はアドレスをＤ０００Ｈあたりでいじってください。間違ってＥ０００Ｈあたりでいじると暴走しますので、気をつけて。

カーソル移動はカーソル・キーを行なって

〈図4－1〉 f･1で選ぶRadio Life etc…モードの画面及びチェックサムの算出方法

ワクの中の合計値の下位1バイト

```
ADRS    +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   SUM  [Ø123456789ABCDEF]
-------------------------------------------------------------------------------------
CØØØ  : CD B5 C3 DD 21 C3 C3 3E:ØØ 32 A7 E6 CD 8B 42 CD ⇒ 2D   |ヘオテン!テテ> 2ァ ヘ B ヘ|
CØ1Ø  : C6 C2 CD B5 C3 3E ØC CD:ØD 3E CD 64 C2 CD A7 C3 :  59   |ニツヘオテ> ヘ >ヘdツヘァテ|
CØ2Ø  : CD 9B C1 CD 46 C1 CD 76:C3 28 26 E6 ØF CD 2C C1 :  ØØ   |ヘ チヘFチヘvテ(& ヘ,チ|
CØ3Ø  : F5 3A D7 C3 B7 2Ø 13 F1:CB 27 CB 27 CB 27 CB 27 :  6C   |時:ラテキ  円ヒ'ヒ'ヒ'ヒ'|
CØ4Ø  : F5 3E ØF A6 C1 8Ø 77 C3:A8 CØ 3E FØ A6 C1 8Ø 18 :  F8   |時> ヲチ wティタ>×ヲチ  |
CØ5Ø  : F5 FE 1C CA A8 CØ FE 1D:CA C9 CØ FE 1E CA 19 C1 :  6F   |時  ハィタ  ハノタ  ハ チ|
CØ6Ø  : FE 1F CA 22 C1 Ø1 ØØ Ø1:FE F8 2Ø Ø9 2A D3 C3 Ø9 :  B4   |  ハ"チ         *モテ |
CØ7Ø  : 22 D3 C3 18 AB FE F9 2Ø:ØA 2A D3 C3 ED 42 22 D3 :  8Ø   |"モテ ォ      *モテ□B"モ|
CØ8Ø  : C3 18 FØ FE FE 2Ø Ø3 C3:ØØ CØ FE FF 2Ø 11 CD B5 :  1D   |テ ×     テ タ     ヘオ|
CØ9Ø  : C3 3E Ø1 32 A7 E6 CD 9Ø:42 3E ØC CD ØD 3E FF CD :  8E   |テ> 2ァ ヘ B> ヘ > ヘ|
CØAØ  : 2C C1 77 3E Ø1 32 D7 C3:3A D7 C3 B7 28 5E 3E ØØ :  BE   |,チw> 2ラテ:ラテキ(^>  |
CØBØ  : 32 D7 C3 3A D5 C3 3D 21:EØ C3 BE 2Ø 56 36 ØØ 21 :  2A   |2ラテ:ユテ=!ニテセ V6 !|
CØCØ  : E1 C3 7E FE ØF 2Ø 4C 18:28 3A D7 C3 B7 2Ø 1A 3E :  DE   |ドテ~   L (:ラテキ  >|
CØDØ  : Ø1 32 D7 C3 21 EØ C3 7E:B7 2Ø 3B 3A D5 C3 3D 77 :  A7   | 2ラテ!ニテ~キ ;:ユテ=w|
CØEØ  : 21 E1 C3 7E B7 2Ø 2F 18:15 3E ØØ 32 D7 C3 C3 2Ø :  63   |!ドテ~キ /   > 2ラテテ |
CØFØ  : CØ 2A D3 C3 ED 4B D5 C3:Ø9 22 D3 C3 18 FØ 2A D3 :  16   |タ*モテ□Kユテ "モテ ×*モ|
-------------------------------------------------------------------------------------
SUM   : Ø6 68 F6 76 Ø5 B7 14 1B 6E BC C6 A6 6A 65 AC 78 ⇒ 1E
```

タテのワクの中の合計値の下位1バイト

タテのチェックサムの合計値の下位1バイト

アドレス　内容

```
Data Address = &HCØØØ : CD
```

カーソルのある位置のアドレスと内容を表示

ください。カーソルが画面の最上位、もしくは最下位にいる場合は、自動的にスクロールされます。大幅に移動したり流して見る時などはＲＯＬＬ UP/ＲＯＬＬ DOWNキーを使うと、256バイトごとに前後移動ができます。

なお、実行中にＦ・１を押すと図３へ、Ｆ・２を押すと終了してモニタへ戻れます。再度の実行はモニタの状態で〝ＧＥ０００〟＋リターンを実行します。(リスト１の時。リスト２なら〝ＧＣ０００〟＋リターン)

みごとに動いたら次はこのHexai／Editerを使ってリスト２を打ち込みましょう。今度はリアルタイムにチェックサムが変化しますから一発入力間違いなしダゼ／

●SAVEのしかた

完成したプログラムはＳＡＶＥしときましょう。

［リスト１］

※ＤＩＳＫ：ＢＳＡＶＥ〝ＨＥＸＡＩ．Ｅ０〟、&ＨＥ０００、&Ｈ５００⇦BASICで実行。

※ＴＡＰＥ：WHEXAIE、Ｅ０００、Ｅ４ＦＦ⇦モニタで実行。

［リスト２］

※ＤＩＳＫ：BSAVE〝HEXAI．Ｃ００〟、&HC000、& H500⇦BACICで実行。

※ＴＡＰＥ：WHEXAIC、Ｃ０００、Ｃ４ＦＦ⇦モニタで実行。

●終わりだぜ

さて、ちょっと長いがとても便利なHexai／Editerを使って皆んなも、もっと機械語に親しみましょう。

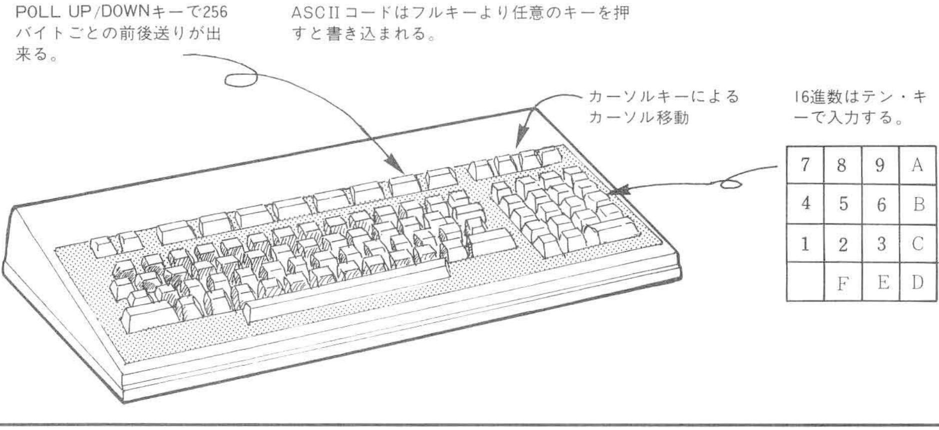
機械語は決してムズカシイものではありません。現に某編集部には小学生が作ったＭＳＸの機械語のゲームが届いており、これを見てびっくらこいたこともあった位で、ちょっと本を見ればめんどうくさいけれども、とても面白い世界をみることができます。

〈図 4 − 2〉 f- 2 で選ぶ ASCⅡモードの画面及びチェックサムの算出方法

```
ADRS    +Ø +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  SUM  [Ø123456789ABCDEF]
---------------------------------------------------------------------------------
CØØØ  : CD B5 C3 DD 21 C3 C3 3E::  Ø7  ワクの中の値の合計値          |ヘオテン!テテ>       |
CØØ8  : ØØ 32 A7 E6 CD 8B 42 CD::  26                                | 2ァ"ヘ B'ヘ          |
CØ1Ø  : C6 C2 CD B5 C3 3E ØC CD::  E4                                |ニツヘオテ> ヘ        |
CØ18  : ØD 3E CD 64 C2 CD A7 C3::  75                                | >ヘdツヘァテ          |
CØ2Ø  : CD 9B C1 CD 46 C1 CD 76::  4Ø                                |ヘ チヘFチヘv         |
CØ28  : C3 28 26 E6 ØF CD 2C C1::  CØ                                |テ(&" ヘ,チ            |
CØ3Ø  : F5 3A D7 C3 B7 2Ø 13 F1::  A4 ← ASCⅡモードでは               |時:ラテキ  円         |
CØ38  : CB 27 CB 27 CB 27 CB 27::  C8    チェックサムがここに来る!    |ヒ'ヒ'ヒ'ヒ'          |
CØ4Ø  : F5 3E ØF A6 C1 8Ø 77 C3::  63                                |時> ヲチ wテ          |
CØ48  : A8 CØ 3E FØ A6 C1 8Ø 18::  95                                |ィタ>×ヲチ            |
CØ5Ø  : F5 FE 1C CA A8 CØ FE 1D::  5C                                |時  ハィタ            |
CØ58  : CA C9 CØ FE 1E CA 19 C1::  13                                |ハノタ  ハ チ         |
CØ6Ø  : FE 1F CA 22 C1 Ø1 ØØ Ø1::  CC                                |   ハ"チ              |
CØ68  : FE F8 2Ø Ø9 2A D3 C3 Ø9::  E8                                |        *モテ         |
CØ7Ø  : 22 D3 C3 18 AB FE F9 2Ø::  92                                |"モテ オ              |
CØ78  : ØA 2A D3 C3 ED 42 22 D3::  EE                                | *モテ□B"モ          |
---------------------------------------------------------------------------------
 SUM  : 74 E4 36 DD FA ØD 7B AØ => 8D  タテのチェックサムの合計値
      タテのワク内の合計値

 Data Address = &HCØØØ : CD
```

〈図 − 5〉 キーの操作



それと、パソコン・ボーイの皆さん!お願いですから、せめて附属のマニュアル位よく読むようにしてください。

編集部にもちょっとマニュアルを見ればすぐわかるようなことを質問してくる人がとても多く、てんてこまいしているのです。(某編集部のことで編集部にはこんな問い合わせは少ないみたいですよ! 編集長)

高い代物なのですから、十分に使いましょう。マニュアルだって、その値段のうちに入っているんですよ。

さて、もうそろそろ終わりなので、ぐちを少々。いやー今回は疲れました。Z80の開発をするのに困ってしまった。モニタのアセンブラはアホらしくて使っていられないし、何でZ80の機械に8080なんてもののアセンブラ、逆アセンブラが付いているんですかね。ホント!!

Hexai! Editerの使い方

○BASICで

　　WIDTH␣80、25：CONSOLE␣0、25

　　CLEAR␣、＆HE000　　リスト1

　　　　　　＆HC000　　リスト2

　　Diskの場合

　　　　BLOAD␣"HEXAI. E00"　　リスト1
　　　　BLOAD␣"HEXAI. C00"　　リスト2
　　　　MON

　　Tapeの場合

　　　　MON
　　　　RHEXAIE　　リスト1
　　　　RHEXAIC　　リスト2

　　GE000　　リスト1

　　GC000　　リスト2

○ASCII形式かRadio Life式か選ぶ（f.1かf.2を押す）

　アドレスの上位3ケタを入力

●メモリ・エディット

○f.1で図3に戻るかf.2で終了。

○終了したらメモリ内容をテープかDiskにSAVEする。

　テープにはモニタのまま "W(スタートアドレス)、(エンドアドレス)"

　ディスクには[CTRL]と[B]を一緒に押してBASICに戻り

　　BSAVE␣"(ファイルネーム)"、&H(スタートアドレス)、&H(長さ)

○もう一度Hexai! Editerに戻るには

　MON

　GE000もしくはGC000

注意○リスト1のプログラムではDF80H〜DFFFH、

　　リスト2ではBF80H〜BFFFHまではいじらないようにしてください。

　　スタックが積まれているため、下手をすると暴走します。

　○本プログラムは自分自身を書き替える為、実行させ、自分自身を見ても

　　プログラム・リストとはチェックサムが異なりまーす。

不思議ですねぇ。CP/Mで組むのは、いちいちシステムを立ち上げるのが面倒だし、そこで苦肉の策として、APPLEのZ80用アセンブラでプログラムを作り、それをRS-232Cを使って88に送って動作させるという方法を取りました。

使いなれたアセンブラなので、とても楽にできましたが、なぜこんなことをせにゃならんのか、頭を抱えてしまいます。こらNECめ! お兄さんは怒っているんだぞ!!

FM-7 TAPE ⇨ TAPE 《丹治佐一》

究極のカセット・バックアップ・ツール

"スルー"なのだ!!



強力なプロテクトをかけたソフトが出るやいなや、それをバックアップするためのツールが出てしまいます。すると、さらに強力なプロテクトをかけたソフトが発売される、というようにソフトとバックアップ・ツールはイタチごっこの感があります。

FM-7のカセット・テープ・プロテクトも同様なことがいえます。通常はBIOS(基本入出力システム)を利用しているわけですが、これでは簡単にバックアップが取れてしまうためにオリジナル?の入出力ルーチンを使ってプロテクトをかけています。

最近のソフトはこのオリジナル・ルーチンのものが多く、当然、フォーマットも転送スピード(ボーレート)も異なり、BIOSを利用したコピー・ツールはお手上げになってしまいます。

究極のバックアップ・ツール〝スルー〟

そこで考えたのがこの〝スルー〟です。この基本的な考えは、2台のデータレコーダを使ってオーディオ的にバックアップするのと同じです。

ただ、この〝スルー〟では、FM-7をその途中に入れFM-7に入ってくる波形をデジタル変換して再び外に出力させています。そのために、きれいな波形の形でバックアップ・テープが作れてしまいます。

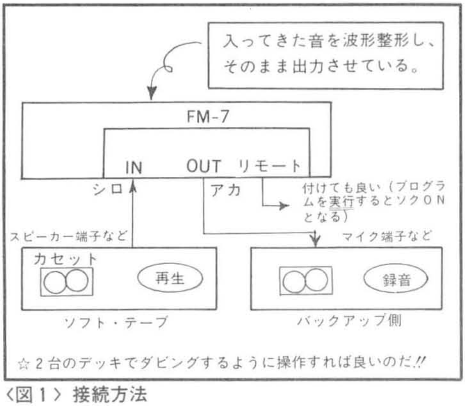
プログラムは、フォーマットもボーレートもまったくチェックしておらず、入ってくる波形情報をそのまま出力しているため、どんな強力なプロテクトがかけにあっても関係なくバックアップしています。

使い方

図1のように2台のデータレコーダの間にFM-7を入れた形で配線します。プログラムをRUNさせると、後は2台のデータレコーダを使ったダビングと同じです。1台のデータレコーダでソフトを再生し、もう1台で録音を開始すれば良いのです。

再生を開始するとFM-7のスピーカから入力された音が出力されますから、録音のタイミングやチェックに利用してください。

プログラムはリスト1に、そのマシン語ソース・リストをリスト2に載せておきます。しかし、このリストを見ても分かる通り、とても簡単にできてしまうのですから、市販の〝2レコーダ用〟と名付けられたコピー・ツールがどんなイイカゲンであるかが、お分かりいただけるでしょう。(リスト1のみ入力)



〈図1〉接続方法

〈リスト1〉

```
100 RESTORE:AD=&H1000
110 READ D$:IF D$="end" THEN EXEC &H1000
120 POKE AD.VAL("&H"+D$):AD=AD+1
130 GOTO 110
140 DATA 20.0A,7D.FD.02.2B,05.CC
150 DATA 43,81,20,03,CC,42,80,B7
160 DATA FD.00,F7,FD.03,7D,03,13
170 DATA 27,E8,86,40,B7,FD,00,39
180 DATA "end"
190 '==================================
200 ' Cassette Bàck Up tool 'Through'
210 '  for FM-7  Copuright 1985.5.15
220 '        bv Saichi-Tanii
230 '==================================
```

〈リスト2〉

```
           ** MC-6809 Assembly Language **

500              (FD02)      READ   EQU   $FD02
510              (FD00)      WRITE  EQU   $FD00
520              (0313)      BRAK   EQU   $313
530              (FD03)      SPEAK  EQU   $FD03
540
550              (1000)             ORG   $1000
560  1000        20 0A              BRA   LOW
570  1002        7D FD02     LOOP   TST   READ
580  1005        2B 05              BMI   LOW
590  1007        CC 4381            LDD   #$4381
600  100A        20 03              BRA   JP
610  100C        CC 4280     LOW    LDD   #$4280
620  100F        B7 FD00     JP     STA   WRITE
630  1012        F7 FD03            STB   SPEAK
640  1015        7D 0313            TST   BRAK
650  1018        27 E8              BEQ   LOOP
660  101A        86 40              LDA   #$40
670  101C        B7 FD00            STA   WRITE
680  101F        39                 RTS
690
700              (101F)             END
```

メモリのアドレス　　マシン語の内容　　ニーモニック（マシン語を作る時に利用したものなので、FM-7に入力する際は不要）

```
           ** Labels & Addresses **

READ     FD02       WRITE    FD00       BRAK     0313
SPEAK    FD03       LOOP     1002
```

編集後記

このところ、各パソコン・メーカーの製品の売れ行きが伸び悩み状態で、かなり商品がダブついているとの噂を秋葉原界隈で耳にしますし、ダンピング価格で最新機種が店頭に並んでいます。

それに比べ、異常とも思えるほどに人気を得ているのが中古販売のお店です。まあ半年の間も置かないうちに、次機種を発売するメーカーもあるわけですから、何かおかしいゾ！とそうやすやすメーカーの思惑にはまらない賢い買い物を求めるユーザーが多くなるのも当然でしょう。

中古として販売されている機種でも、決して最新機種に劣らない性能、機能を持っていますし、発売されて日がたっているためパソコン各誌にありとあらゆるプログラムや解析記事が紹介されているという、とっても有難いメリットがあります。

さらに販売価格が½程度なのですから、人気を集めるのは当然のことといえます。これなら幾ら新しい機種が出ようとも気に病むことはないでしょう。

この先、パソコン業界がどのような道をたどるかはまったく見当もつきませんが、ただ確実なのは，新製品乱発のメーカー主導型の発想からユーザー主導の発想へと変わらない限り我々パソコン・ユーザーにそっぽを向かれてしまうということです。

この件でメーカーに恨みを持っている、いちパソコン・ユーザーとして各メーカーに肝に命じて欲しいと思うのですが……。（福場）

●表紙制作協力：丹治　恵
●イラスト：酒井祖美

●ラジオライフ別冊
バックアップ活用テクニック②

定　価　980円
送　料　250円
発行年月日　昭和60年11月15日第三刷

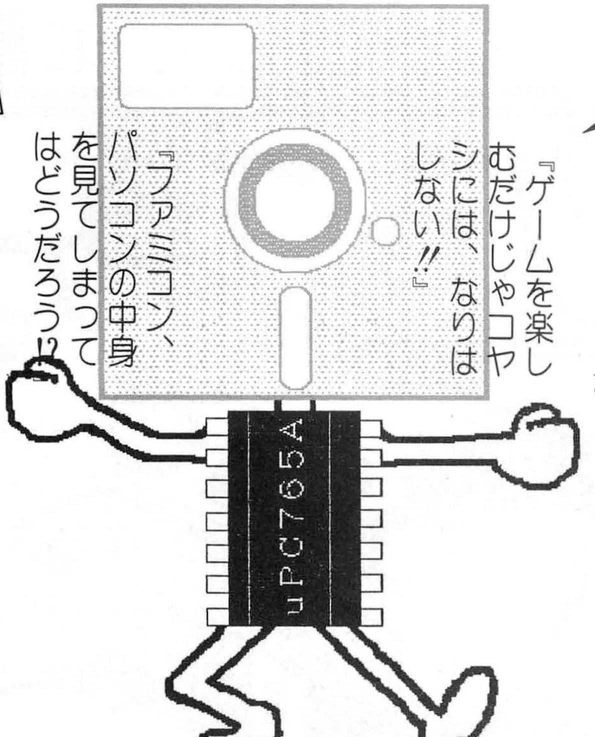
発行所　株式会社　三才ブックス
〒104 東京都中央区八丁堀3-22-9 石橋ビル
電話(03)553-7331(代)

発行人　和田洋一
編集人　福場龍夫
振　替　東京3-58044

印刷所　共同印刷株式会社
写　植　株式会社 サトウ印書館

バックアップは

新 頭のコヤシ!!



バックアップ活用テクニック パート3

3月25日発売!!

ファミコン
バックアップ・テクニック＆大改造法・特集

●日電PC-8801、富士通FM-7/77AV、シャープX1シリーズ、各社MSXのディスク・コピーの実戦テク＆ツール・プログラムを一挙掲載。さらに、GAMEファン待望のファミコン・バックアップ・テク＆初公開!!内部解析データを満載したハッカー必読書。

主な内容

●ディスク・コピーテクニック／ＰＣ、ＦＭ各機種＋Ｘ１シリーズ●ファミコン徹底解析テクニック／ＰＣ88タコ吸いVer.２、ＦＭ・アップル版、大改造法(回路図初公開)●ノウハウ集／77ＡＶコピー法、ＭＳＸコピー法他。

●Ａ５判　定価980円(〒250)

ラジオライフ

誰でも受信できるデータと情報

の月刊誌



●定価五〇〇円 送料六五円

ラジオ1台で楽しめる警察無線、消防無線、救急無線、航空無線、軍用無線、海外放送、パーソナル無線の聞き方・使い方を大特集。情報無線㊙受信法とは。

●電波をフォーカスした話題の月刊誌!!

毎月25日全国書店発売中

書店にない場合は直接小社まで御注文ください。

三才ブックス振替口座No.（東京）3-58044

フォーマット解析＆コピープログラム No.2

PC-8800シリーズ Tape版

- Super COPY MAKER …… ¥4,000
- COPY-BOY ……（1or2レコーダー対応、特殊セーブにも対応、100KB以上を収容）¥4,000
- 名探偵88 ……（3種類のフォーマット指定可）…… ¥3,500
- Magic Copy II（ファイラー別売）…… ¥4,500
- 必殺複写人 …… ¥3,500

PC-6000シリーズ Tape版

- 名探偵60＆mkII（テープのコピー、スタートアドレス、エンドアドレス表示可）¥3,500
- 必殺複写人 …… ¥3,500
- 浪人生 ……（テープ版をディスク版へ。ファイラーサポート付）…… ¥3,800

FM-7／77用 Tape版

- COPY-BOY ……（1レコーダ／2レコーダ 対応 1KBごとに画面表示可）¥3,600
- CHACER-7 ……（64KまでOK。ビットマシンに変化？）…… ¥3,800
- 必殺複写人 …… ¥3,500

シャープ X1 MZシリーズ用 Tape版

- Dr、COPY×1（個別ソフトハドソン対応）…… ¥6,000
- 必殺複写人 …… ¥3,500
- レプリカ（MZ-1500用 QD版）…… ¥5,800
- 必殺複写人（MZ-700用）…… ¥3,500

MSX用 Tape版

- Super COPY MAKER …… ¥4,000
- 必殺複写人 …… ¥3,500
- COPY-JACK …（ROM版をTape版へ Tape to Tapeもできます）…… ¥6,800
- ロリコピーMSX（ROMのゲームをTape版に。Tape to Tapeも出来ます。）¥6,8[illegible]

DISK・SAVER

- DISK・SAVER-98F（1枚のディスクに5～10本ぐらいのプログラムをSAVEできます。）¥6,800
- SWEET君 …（テープ版をディスク版へ SWEET88が必要）¥3,800
- N.S君 ……（テープ版をディスク版へNS-DOSが必要）… ¥3,800
- DISK・SAVER-88（SWEET-88が必要・プロテクトがかけてあるものでもOK。テープ版のソフトをディスク版に）¥6,800

GAME-DOS

（GAME-DOSとはROMモードでディスク・システムを使用可能にするDOSです）

（GAME-DOSをお買上げの方にGAME-DOSマニアルをサービス）

PC-8800シリーズ用

- NS-DOS …… ¥9,800
- C-DOS …… ¥9,800
- SWEET88 …… ¥9,800
- T88-DOS …… ¥9,800

FM-7用

- FIXER-7（プログラムをF-BASICのファイルに変換。プロテクトがかけてあってもOK。ファイラーはすべてサポート）…… ¥11,800
- 留年生 …（テープ版のソフトをディスク版へ変換。プロテクトがかけてあってもOK。30種以上のサポート）¥ 9,800

★その他多数有りますので一度ご来店ください★

新作ゲームソフト 大量入荷中！ お問合わせ下さい。

株式会社 若松通商

本社 〒101 東京都千代田区外神田1-11-4 ミツワビル 1・2階 ☎03(251)4121代 FAX.03-251-7877(GII・GIII)

マイコンセンター若松 〒101 東京都千代田区外神田1-11-4 ミツワビル BF ☎03(251)8933

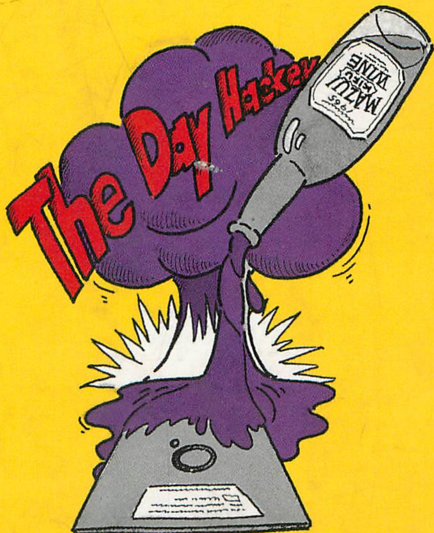
秋葉原店 〒101 東京都千代田区外神田1-15-16 秋葉原ラジオ会館4階 ☎03(255)5064

通販部 〒211 神奈川県川崎市中原区小杉陣屋町1-547-80 中原郵便局私書箱37号 ☎044(722)0948 ☎03(257)0063 FAX.044-733-7345

マイコンパーツセンター 〒101 東京都千代田区外神田1-9-9 内田ビル3階 ☎03(253)8521代

送料サービス券

森本



森本

三才ブックス　〒104 東京都中央区八丁堀3-22-9 石橋ビル ☎03-553-7331(代)　定価980円　雑誌09156-11/15　第三刷